CONTENT-BASED ANALYSIS OF DIGITAL VIDEO

Alan Hanjalić

Kluwer Academic Publishers

CONTENT-BASED ANALYSIS OF DIGITAL VIDEO

CONTENT-BASED ANALYSIS OF DIGITAL VIDEO

Alan Hanjalić

Delft University of Technology
Delft, The Netherlands

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 1-4020-8115-4
Print ISBN: 1-4020-8114-6

Created in the United States of America

Visit Springer's eBookstore at: http://www.ebooks.kluweronline.com
and the Springer Global Website Online at: http://www.springeronline.com

To my family

Contents

Preface

This book explores the underlying principles, concepts and techniques of *video content analysis,* an exciting and rapidly growing research area with great potential of causing paradigm shifts in traditional educational, professional, business, communication and entertainment processes. This great potential emerges from broadening the scope of possibilities of handling video. Essentially, the aim is to develop systems that are capable of "understanding" video, or in other words, that are capable of extracting information about the content conveyed by the "meaningless" zeros and ones of the digital video data stream. Examples are automated video-recommender systems for personalized delivery of television broadcasts in our homes, the systems providing easy access to the content of vast film and video archives at broadcasters, museums, industries and production houses, automated video surveillance systems, the systems enhancing video-based business communication and, finally, the systems revolutionizing remote education and e-learning.

Video content analysis is a strongly multidisciplinary research area. The need for a multidisciplinary approach becomes clear if one realizes that under the term "digital video" we understand, generally, a multimodal data stream. This stream typically consists of a visual, audio and text component each of which carries a part of the content information. Clearly, powerful techniques for information extraction from images, sounds, speech and text are required to successfully meet the video interpretation challenge. For this purpose, the knowledge from the areas of image and audio processing, speech recognition, "classical" (text) information retrieval and natural language processing can be applied. Further, because the task of content interpretation can be seen as a task of assigning the "chunks" of digital video

data to content categories, such as “Alpine landscape”, “Sport highlight”, or “Excitement”, on the basis of characteristic “patterns” found in the data, the realization of this task can be approached using the theory and techniques of pattern classification. Finally, psychology plays an important role in the process of interpreting video at the affective level, that is, of identifying the video segments that are “exciting”, “sad” or “happy”.

The topics treated in this book are critical for successfully approaching the challenge described above. These topics include

- Video parsing into shots and semantic content segments,
- Video content indexing and representation for browsing and retrieval,
- Affective video content analysis for mood extraction and personalization of video content delivery.

The first topic considers the problems of low-level and high-level video parsing. While low-level parsing stands for the detection of elementary temporal segments in video, or *shots,* high-level parsing searches for the boundaries between *semantic content segments,* which can be seen as shot aggregates characterized by a coherent story line. The parsing results provide the basis for the processes belonging to the second topic listed above, namely, automatically indexing temporal segments of video according to prespecified content labels. By applying algorithms for automated indexing of video content and its organization in user-friendly content browsing and retrieval schemes, the segments corresponding to “action”, “dialog”, “goal in a soccer match” or “hunt scene in a wildlife video” can be made easily accessible to the user. The third topic – affective video content analysis – addresses the theory that aims to extend the possibilities for content access from the cognitive to the affective level. With the objective of recognizing emotions and moods that are communicated by a video toward the user, affective video content analysis is likely to enable the development of powerful tools for many new application contexts, such as for personalizing the delivery of video content toward the user.

In the way that is typical for a textbook, this book integrates the existing, rather fragmented knowledge related to the above topics into a unified and fundamental theoretical approach that may serve as a guide and inspiration for conducting further research on video and multimedia content analysis and retrieval. I wish you a pleasant reading.

Alan Hanjalić Delft, April 2004

Acknowledgments

I would like to use this opportunity to express my gratitude to a number of colleagues for their help and support in the development of this book.

First of all, I thank Professor Jan Biemond and Ir. Richard den Hollander from Delft University of Technology for valuable suggestions, which greatly improved the quality of the material presented in the book.

Special thanks go to Professor Rosalind W. Picard from Massachusetts Institute of Technology (MIT) Media Laboratory and to Professor Annie Lang from Indiana University for reviewing the "affective" Chapter 5 of the book. The material presented in this chapter has emerged from my research on affective video content analysis that I started during my stay at British Telecom Labs (BTExact) in 2001. I thank Dr. Li-Qun Xu for inviting me to BT and for a very pleasant and fruitful cooperation in the past years.

I also thank Kluwer Academic Publishers, particularly Alex Greene and Melissa Sullivan, for making this book possible.

Chapter 1

INTRODUCTION

1.1 VIDEO CONTENT ANALYSIS: TOWARD A PARADIGM SHIFT

Recent advances in video compression technology, the availability of affordable digital cameras, high-capacity digital storage media and systems, as well as growing accessibility to Internet and broadband communication networks have led to vast popularity of digital video. Not only that digital video increasingly replaces analog video in various application contexts, but also the amount of digital video being produced, watched, edited, stored, broadcasted and exchanged is already phenomenal and quickly growing.

The development described above is enabled by the possibility to process digital video automatically, either for the purpose of compression, editing, streaming through a network, or simply for displaying it on a portable electronic device. The possibilities to process digital video reach, however, far beyond these tasks. In particular, we could process digital video data with the objective of extracting the information about the content conveyed by this data. The algorithms developed for this purpose, referred to as *video content analysis* algorithms, could serve as the basis for developing the tools that would enable us, for instance, to easily access the events, persons and objects captured by the camera, or to efficiently generate overviews, summaries and abstracts of large video documents. Figure 1-1 illustrates the benefits of video content analysis on the example of an algorithm set capable of recognizing the "chunks" of digital video data showing an "Alpine landscape", a "news report on topic *T*", the "highlights of a soccer match" or some "suspicious behavior" detected in a surveillance video, and of making this selected content easily accessible to the user.

The availability of algorithms for video content analysis may lead to a radical paradigm shift in the traditional educational, professional, business, communication and entertainment processes. In the following we illustrate the possibilities and consequences of such a shift on a number of typical application scenarios.

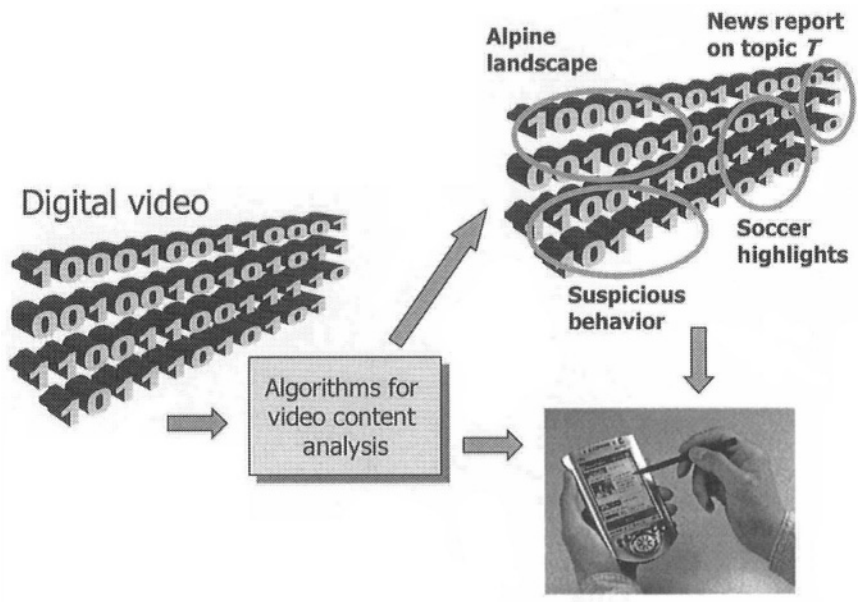

Figure 1-1. Algorithms for video content analysis can be developed to reveal the content conveyed by digital video data and to make this content easily accessible to the user.

1.1.1 Video broadcasting

With the advent of digital video revolution the television broadcasting industry is slowly but surely transferring to an end-to-end digital television production, transmission and delivery chain. Supported by the availability of broadband communication channels, this transfer will lead to an enormous increase in the amount of video data reaching our homes. At the same time the quickly growing capacity-versus-price ratio of digital storage devices is likely to make such devices highly popular with consumers. A combination of the abovementioned phenomena will result in an explosion in the "consumer choice", that is, in the number of video hours that are instantaneously accessible to the consumer. This may have crucial consequences for the ways the broadcasted material is "consumed". As

reported in the study by Durlacher Research Ltd. [Whi00], the understanding of the broadcasting mechanism may change. This mechanism will only be something that provides data to the - soon inevitable - *home mass storage system* (HMSS) and, as far as the consumer is concerned, the concept of the "broadcasting channel" will lose its meaning. Further, due to the large amounts of incoming data, video recording will be performed routinely and automatically, and programs will be accessed on-demand from the local storage. Viewing of live TV is therefore likely to drastically diminish with the time [Whi00].

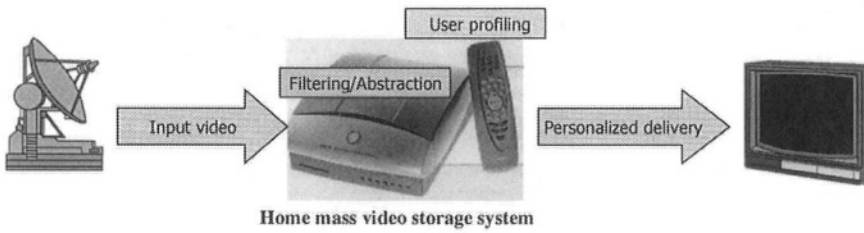

Figure 1-2. Two major modules of the video-recommender functionality embedded into a home mass storage system are the video abstraction and personalization modules. On the basis of the user profile, the incoming video content is filtered, organized and presented to the user via abstracts.

The challenge of securing the maximum transparency of the recorded video volume toward the consumer - independent of the volume size – could be approached by developing *video-recommender* functionality of a home mass storage system. As indicated in Figure 1-2, this functionality would typically contain the following two main algorithmic modules:

- Module for automatically abstracting video,
- Module for matching the incoming video material with user preferences.

The purpose of an algorithm for video abstraction in the context of a video-recommender functionality can be twofold. First, the abstraction algorithm can be designed to *summarize* the broadcasted material in order to facilitate the consumer's choice of what to watch later on. This may be highly valuable, for instance, in the process of digesting a large volume of news television broadcasts and presenting to the user in a compact but

comprehensive way the coverage of all news topics found in the volume. Alternatively, a video abstraction algorithm can be designed to *prune* the recorded video material by keeping the most interesting segments – *highlights* - only, and by discarding the remaining, less interesting parts. For instance, pruning is particularly applicable to sport broadcasts as the events being worth watching (e.g. goals in soccer, home runs in baseball, touchdowns in football) are sparse and spread over a long period of time.

The second module, also referred to as the *personalized video delivery* module, addresses the problem of storing and organizing the incoming video material according to the subjective preferences of the consumer. Ideally, these preferences are stored in the *user profile* that is acquired in a *non-invasive* fashion, that is, without requesting complicated or uncomfortable actions from the consumer. The systems currently available for personalized video delivery typically process a video on the basis of textual information like, in the case of a movie, the genre, cast, director and script. As the user profile - particularly in the case of a movie - is also largely determined by the prevailing mood of a movie, then the information about the mood is likely to enhance the personalization process. For this purpose, an algorithm could be developed that analyzes the types and intensities of emotions and moods along a movie, infers the changes in the prevailing mood and then matches the mood of a particular part of a movie with the current or preferred mood of the consumer.

1.1.2 Video archives

Vast film and video archives exist at broadcasters, museums, industrial organizations and production houses worldwide, documenting the world's cultural, social and political development over the past century, and thus representing the irreplaceable record of our heritage. With quickly developing digital technology the tendency grows toward transferring the content of film and video archives into a digital format. This tendency has a twofold origin. First, the audiovisual material stored in archives was created on a wide range of equipment, much of which has become obsolete and difficult to keep operational. Digitizing the entire material would then transfer it into a uniform format, viewable on modern digital displaying equipment. The second reason for digitizing traditional archives is the "physical" content protection. In many cases the original film- and videotapes have degraded to the point where the defects due to age and wear make the replaying quality quite unacceptable. Transferring a film or a video into a digital format makes it possible to archive it on modern digital storage media and to restore its quality using the techniques of image, audio and speech analysis and processing (filtering).

Once an archive has been digitized, tools for video content analysis may be used to automatically index the archived material in a content-based way. This is an efficient solution for making this material easily accessible to a broad public: many hours of manual work per each hour of video would be needed otherwise to perform the required content analysis and annotation steps. Indexed archives can then be connected to digital transmission networks to make the unique audiovisual records of the past widely available for informative, educational and entertainment purposes.

1.1.3 Security

The demand for more public safety directly translates to the need for more extensive surveillance of public places - a demand that can be fulfilled only by drastically increasing the number of surveillance cameras installed at relevant locations. Needless to say, the amount of surveillance videos collected from all these cameras is much larger than what an affordable number of surveillance officers can analyze to evaluate the safety at a given place over time.

The scalability problem described above can be solved by using intelligent surveillance cameras, which are equipped with video content analysis algorithms. These cameras could detect and recognize suspicious events autonomously, alert the authorities directly and, in this way, strongly reduce the need for additional surveillance personnel and related costs.

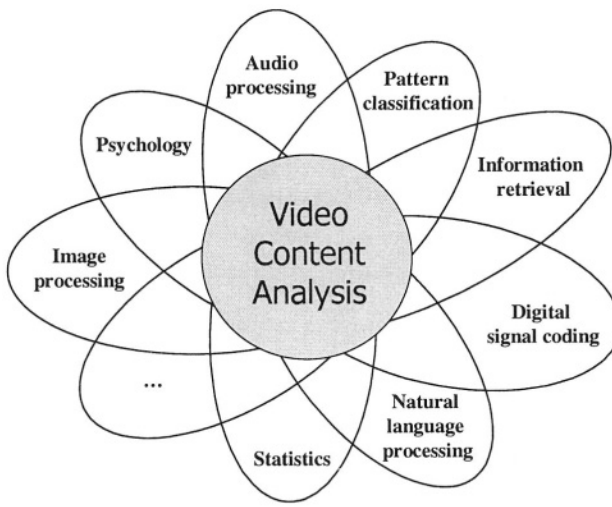

Figure 1-3. The research area of video content analysis benefits from contributions originating from a large number of different research fields

1.1.4 Business and education

By providing the users with the relevant content information about the videos stored on the Internet, video content analysis algorithms can make the Internet video assets better manageable, searchable and reusable. As the same is applicable for large remote instructional video archives as well, the employment of video content analysis algorithms is likely to revolutionize the remote education and *e-learning.* A typical type of content information to be generated for an Internet or instructional video is a concise but comprehensive summary of all topics or thematic units found in a video. By only downloading these summaries, the user can search for the topic or thematic unit of interest with maximized efficiency of interaction with the video collection. Finally, the tools mentioned above can also be used to record a meeting and to automatically generate the minutes and a video summary of the main points of the meeting.

1.2 TOWARD THE MEANING OF VIDEO DATA: BRIDGING THE SEMANTIC GAP

Triggered by the great new possibilities, some of which we outlined in the previous section, a new multidisciplinary research area has emerged worldwide in the late eighties having as the main objective the development of algorithms for video content analysis. As illustrated in Figure 1-3, image, audio and speech signal processing, psychology, pattern classification, information retrieval, digital signal coding, natural language processing and statistics are only some examples of many research fields contributing to this area.

The scientific challenge when developing video content analysis algorithms is to "bridge the semantic gap" (Figure 1-4), that is, to infer the content conveyed by the given data set from the representation of that set available in the form of *low-level features.* Formally, the semantic gap can be defined as follows [Sme00]:

Definition 1.1

Semantic gap is the lack of coincidence between the information that one can extract from the digital data and the interpretation that the same data has for a user in a given situation.

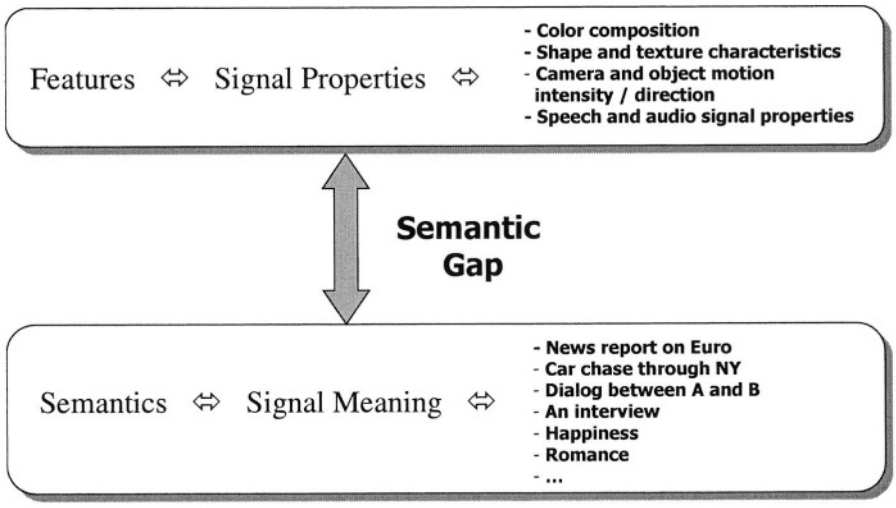

Figure 1-4. Semantic gap: The gap between the content conveyed by the data and the low-level properties of the data

Under the content conveyed by the data we can understand, for instance, the scenes, events, persons, moods or story contexts that one would actually perceive when watching a video, that would make one watch a video in the first place or that would make one remember a video. Low-level features (further on also referred to as *features*) are the results of measurements performed on the data. Examples of features used in the practice of video content analysis are

- *color features* (e.g. color distribution and color moments),
- *texture features* (e.g. textural energy, contrast, coarseness, directionality, spectral frequency coefficients, wavelet coefficients, repetitiveness, complexity, auto-correlation, co-occurrence matrix, fractal dimension, auto-regressive models, stochastic models),
- *shape features* (e.g. edge statistics, curvature parameters),
- *audio and speech features* (e.g. pitch, frequency spectrum, temporal signal characteristics, zero-crossing rate, phonemes),

- *motion features* (e.g. motion direction and intensity, motion field coherence),

- *relational features* (e.g. directional and topological relationships between lines, regions or objects).

As already indicated by the Definition 1.1, the content of a given piece of video is not unique and can be perceived in many different ways. Clearly, each way of perceiving video content requires a particular type of information in order to index, classify, filter or organize the video collection correspondingly. As depicted in Figure 1-5, we differentiate between two basic levels of video content perception, hence two different levels of analyzing video content:

- Cognitive level,
- Affective level.

An algorithm analyzing a video at cognitive level aims at extracting information that describes the "facts", e.g., the structure of the story, the composition of a scene, and the objects and people captured by the camera. For example, these facts can be represented by the labels such as "a panorama of San Francisco", an "outdoor" or "indoor" scene, a broadcast news report on "Topic *T*", a "dialog between person A and person B", or the "fast breaks", "steals" and "scores" of a basketball match.

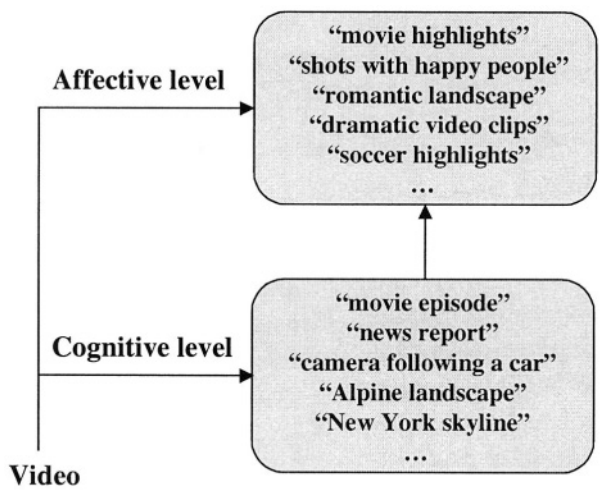

Figure 1-5. Overview of two different levels of video content perception, analysis and retrieval

Affective video content analysis aims at recognizing *affect* (e.g. emotions and moods) in audiovisual data. Typical examples are searching for video segments characterized by "happiness", identifying "sad" movie fragments and looking for the "romantic landscapes", "sentimental" movie segments, "movie highlights" or the "most exciting" moments of a sport event. While an algorithm for extracting affective content could be applied to a video independent of the algorithms operating at cognitive level, affective video content analysis may also be employed to refine the results obtained by analyzing the cognitive aspects of the video content. For instance, assuming that a cognitive analysis algorithm was used to find video clips showing San Francisco, re-analyzing these clips at affective level could define subsets of clips corresponding to "romantic", "sad" or "most memorable" views on San Francisco. Further, in view of the discussion in Section 1.1.1, the affective video content analysis may provide means for enhancing the personalization module of the recommender functionality of a home mass storage system.

We can define three general groups of video content analysis algorithms, namely

- **Video parsing,** which stands for temporal segmentation of a video data stream. It is important to distinguish between *low-level parsing,* where video is segmented in elementary temporal units, or *shots,* and *high-level parsing,* where boundaries between shot aggregates (e.g. movie episodes or news reports) are detected.

- **Video content indexing,** which stands for automatically assigning "chunks" of video data to prespecified semantic categories. The links to these categories are established by means of semantic labels, or *indexes,* (e.g. "Alpine landscape" or "Happiness") using which the corresponding video segments can be retrieved later on.

- **Video content abstraction and representation,** which stands for building compact but comprehensive abstracts of the video segments, which were extracted and indexed using the algorithms described above. The goal here is to efficiently and effectively communicate the essence of the content of these segments toward the user.

Clearly, the first group of algorithms aims at detecting the building blocks of the video content structure, while the algorithms from the second group aim to identify the blocks with a particular content. The third group of algorithms build on the results of the parsing and indexing steps and make the content of the extracted and indexed blocks accessible to the user.

Independent of the type of the algorithm for video content analysis that is to be developed, a number of crucial issues need to be taken into account during the process of algorithm development, such as

- **Level of automation:**

  In order to reach the optimal level of interactivity for the user, a tool for video content analysis should ideally allow only for as much interactivity as really necessary in a given application context. For instance, while a surveillance video system should work fully automatically, certain freedom should be left to the user of a consumer video storage device to determine the length of the extracted movie summary or sport highlight. Then, the level of automation obtained through the analysis of application requirements can serve as a fixed parameter in subsequent steps of optimizing the algorithm performance and efficiency.

- **Multi-modal information fusion:**

  We generally refer to video as a composite signal the parts of which belong to several different modalities. The main modality is the visual one (picture) and it can be accompanied by an audio stream and text captions. The audio stream can further consist of music, speech, environmental noise or any combination of the three. If different modalities are present, then the video content is likely to be a function of information conveyed by all of these modalities. Then, a multi-modal approach to content extraction is necessary, which is based on the fusion of the information derived from different modalities. This is, however, not always the case. There are various examples of video genres where the information conveyed by certain modalities does not contribute to knowledge inference, and may even be misleading.

- **Efficiency and Robustness:**

  In order to be suitable for implementation in the tools used in practical applications, video content analysis algorithms need to be efficient and robust. While the efficiency equals to the speed of the content extraction process, we define the robustness as the capability of a content analysis algorithm to perform reliably in the entire application scope for which it is developed. For instance, an algorithm for extracting topics from news broadcasts should perform equally well for any news broadcast.

1.3 BOOK OBJECTIVE, SCOPE AND OVERVIEW

The objective of this book is to integrate the existing, rather fragmented knowledge related to various issues in the area of video content analysis into a unified and fundamental theoretical approach that may serve as a guide and inspiration for conducting further research in the area. The book covers both the cognitive and affective aspects of video content analysis, with an implicit treatment of the issues of automation, multi-modal information fusion, efficiency and robustness.

The material presented in the book is organized in a bottom-up fashion. In Chapters 2 and 3 we elaborate on the possibilities for revealing the temporal content structure of a video document. Once this structure is known, the concepts discussed in Chapter 4 and 5 can be applied to discover the content in the building blocks of the structure. Although Chapter 4 and Chapter 5 both address the problem of video content indexing, we treat its cognitive (Chapter 4) and affective (Chapter 5) aspects separately. Furthermore, Chapter 4 also treats the problem of video content abstraction and representation for retrieval. In the following we give a brief overview of the content of each chapter.

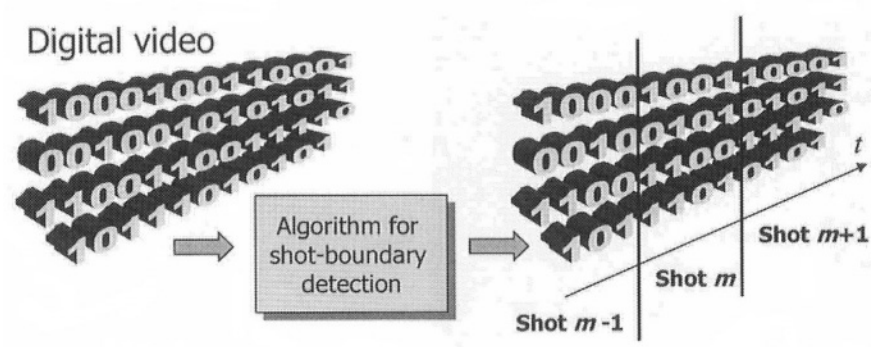

Figure 1-6. An illustration of the objective of shot-boundary detection

1.3.1 Overview of Chapter 2

An elementary temporal unit of a video – a *shot* – can be defined as a series of interrelated consecutive frames taken contiguously by a single camera and representing a continuous action in time. Through parsing a video into shots, the basis is created for a large majority of the video content analysis algorithms. This parsing is done by detecting the boundaries that

separate consecutive shots (Figure 1-6). Shot boundaries can either be abrupt (cuts) or gradual (e.g. dissolves, fades or wipes).

Chapter 2 starts by unraveling the shot-boundary detection problem and by identifying major issues that need to be considered for securing robust detection performance. The attribute "robust" is related here to a constant excellent performance while operating in the "black-box" fashion, that is, unsupervised and self-adjusting. Then, we develop a theoretical framework for solving the shot-boundary detection problem in a general case, which takes into account all issues identified in the beginning of the chapter, and which is based on the Bayesian decision theory.

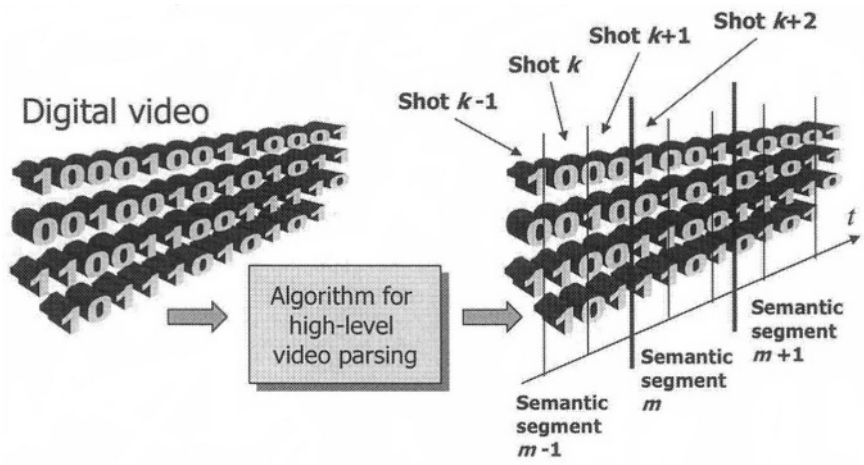

Figure 1-7. An illustration of the objective of high-level video parsing

1.3.2 Overview of Chapter 3

In this chapter we address the problem of high-level video parsing. This is a content analysis step that is typical for video genres characterized by a clearly sequential content structure. A video belonging to these genres can be modeled as a concatenation of separate contexts - *semantic segments* - each of which is potentially interesting for retrieval. The aim of an algorithm for high-level video parsing is then to detect the boundaries between consecutive semantic segments. Examples of such segments are the reports in a broadcast news program, the episodes in movies, the topic segments of documentary programs, or the scenes in a situation comedy.

A semantic segment can be seen as a series of video shots that are related to each other with respect to their content. We therefore approach the problem of high-level video parsing by investigating the coherence of the

content along the neighboring video shots, and search for semantic segment boundaries at the time stamps characterized by sufficiently low content-coherence values. The objective of high-level video parsing is illustrated in Figure 1-7.

Chapter 3 starts by explaining the principle of content coherence, and by introducing the notions of *computable content coherence* and *parsable video.* Then, it is discussed how the content-coherence principle can be applied in practice and used as the basis for developing robust high-level video parsing algorithms.

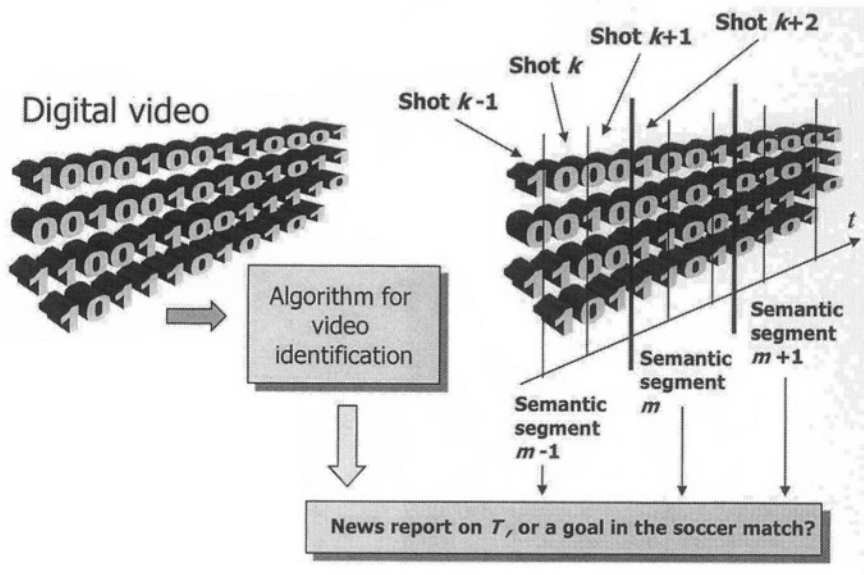

Figure 1-8. An illustration of the idea of video indexing

1.3.3 Overview of Chapter 4

Subsequent to the parsing steps discussed in the previous two chapters, two additional steps are required to provide a fast and easy access to the parsed video material. The first step employs *video indexing algorithms* that recognize in a video the segments belonging to prespecified content classes (Figure 1-8). These classes usually reveal the interest of the user and can be defined as, for instance, the fast breaks, steals or scores in a basketball match, the goals and goal chances in a soccer game, the news reports on a specific topic, the dialogs, actions and story units in a movie or, simply, the hunt scenes in a wildlife documentary.

In the second step, the parsed and indexed video material is brought to the user by means of *video abstracts.* An abstract of a temporal video segment can be seen as a sequence of video frames, possibly with accompanying audio stream, that shows the essence of the content of that segment in a compact but comprehensive fashion. Video abstract can be used to quickly browse through preselected video documents in the search for the documents of interest.

Video indexing will be introduced in Chapter 4 as a pattern classification problem. The treatment of this problem will concentrate on fundamental issues that need to be taken into account when developing robust video indexing algorithms, and on general classes of pattern classification tools that can be used for this purpose. Then the possibilities for video abstract generation will be discussed with the emphasis on compactness, and comprehensiveness of abstract structure.

1.3.4 Overview of Chapter 5

We conclude this book by looking into a relatively new research direction in the area of video content analysis - the representation and modeling of *affective video content.* The affective content of a given temporal video segment can be defined as the intensity and type of *affect* (emotion, mood) that is expected to arise in the user while watching that video segment. The availability of methodologies for automatically extracting this type of video content will extend the current scope of possibilities for video indexing and retrieval. For instance, we will be able to search for the funniest or the most thrilling parts of a movie, or the most exciting events of a sport program (Figure 1-9). Further, as the user may want to select a movie not only based on its genre, cast, director and story content, but also on its prevailing mood, the affective content analysis is also likely to enhance the quality of personalizing the video delivery to the user.

We present in this chapter a computational framework for affective video content representation and modeling. This framework is based on the *dimensional approach to affect* that is known from the field of psychophysiology. According to this approach, the affective video content can be represented as a set of points in the *2D affect space,* which is characterized by the dimensions of *arousal* (intensity of affect) and *valence* (type of affect). We map the affective video content onto the 2D affect space by using the models that link the arousal and valence dimensions to low-level features extracted from video data. This results in the arousal and valence time curves that, when considered either separately or combined into the so-called *affect curve,* are introduced as reliable representations of

expected transitions from one affective state to another along a video, as perceived by a viewer.

The chapter is completed by an overview of possible applications of affective video content analysis. In particular, video indexing using affective labels, highlights extraction and personalized video delivery are addressed.

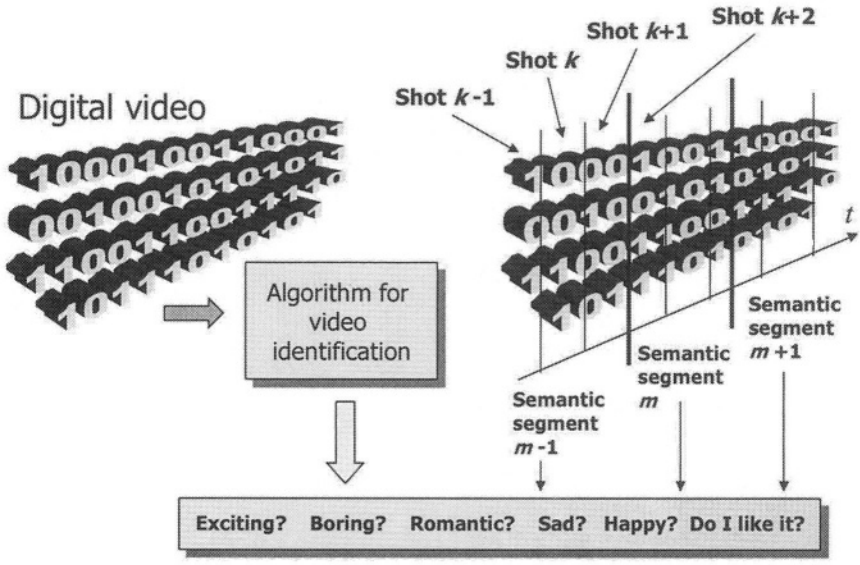

Figure 1-9. An illustration of the idea of affective video content analysis

1.4 REFERENCES

[Sme00] Smeulders A.W.M., Worring M, Santini S., Gupta A., Jain R.: *Content-based image retrieval at the end of the early years,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 12, December 2000

[Whi00] Whittingham J.: *Digital Local Storage: PVR's, Home media servers, and the future of broadcasting,* report by Durlacher Research Ltd., November 2000

Chapter 2

DETECTING SHOT BOUNDARIES IN VIDEO

2.1 INTRODUCTION

Parsing a video into its basic temporal units – *shots* – is considered the initial step in the process of video content analysis. A shot is a series of video frames taken by one camera, like, for instance, by zooming in on a person or an object, or simply by panning along a landscape. Two consecutive shots are separated by a *shot boundary* that can be either *abrupt* or *gradual.* While an abrupt shot boundary, or a *cut,* is generated by simply attaching one shot to another without modifying them, a gradual shot boundary is the result of applying an *editing effect* to merge two shots. Although these effects are numerous, three of them are used most frequently:

- Dissolve,
- Fades (fade-in and fade-out),
- Wipe.

Figure 2-1 shows an example of a dissolve and a cut appearing one after each other in a segment of a TV news program. The dissolve effect is generated by making the first shot darken progressively (fade-out) while at the same time letting the second shot gradually emerge from black to light (fade-in, illustrated in Figure 2-2). As a result, the frames belonging to a dissolve (e.g. frames 4-8 in Figure 2-1) contain a combination of the visual material originating from both shots. Note that a fade-in of the second shot can also simply follow a fade-out of the first shot without superimposing the visual material in modified shot segments. We refer to this effect as *fade group.* Finally, the wipe effect involves a line or a pattern that separates the

visual material of two shots, and that moves across the frame enabling the second shot to replace the first one. An illustration of a wipe is given in Figure 2-3.

The development of shot-boundary detection algorithms has the longest and richest history in the area of video content analysis. Longest, because this area was actually initiated by the attempts to automate the detection of cuts in video, and richest, because a vast majority of all works published in this area so far address in one way or another the problem of shot-boundary detection. This is not surprising since detection of shot boundaries provides the base for nearly all high-level video content analysis approaches, and is, therefore, one of the major prerequisites for successfully revealing the video content structure. Moreover, other research areas can also benefit from automating the shot-boundary detection process. For instance, the efficiency of video restoration can be improved by comparing each shot with previous ones and – if a similar shot in terms of visual characteristics is found in the past – by adopting the restoration settings already used before. Also, in the process of coloring black-and-white movies the knowledge about shot boundaries provides time stamps where a switch to a different gray-to-color look-up table should take place.

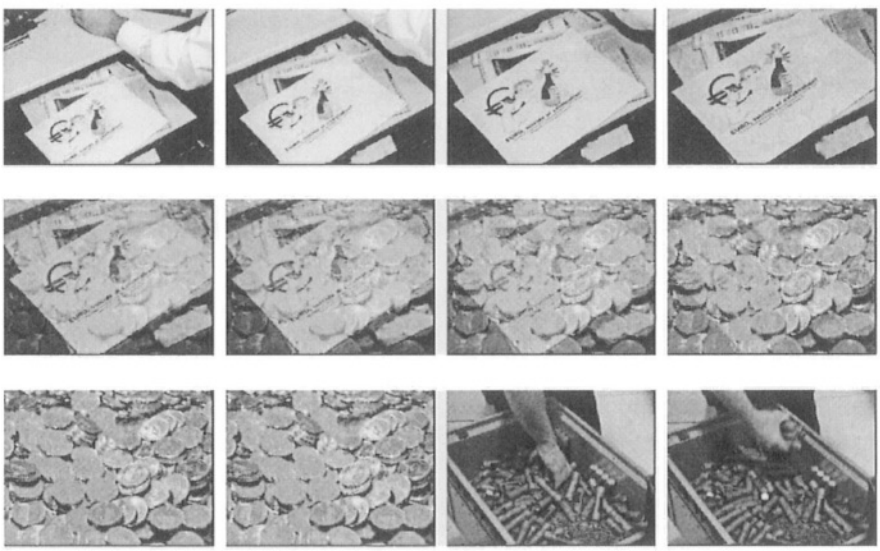

Figure 2-1. Three TV News shots. First two shots are separated by a dissolve effect (Frame 4-8). The second and the third shot are separated by a cut surrounded by frames 10 and 11.

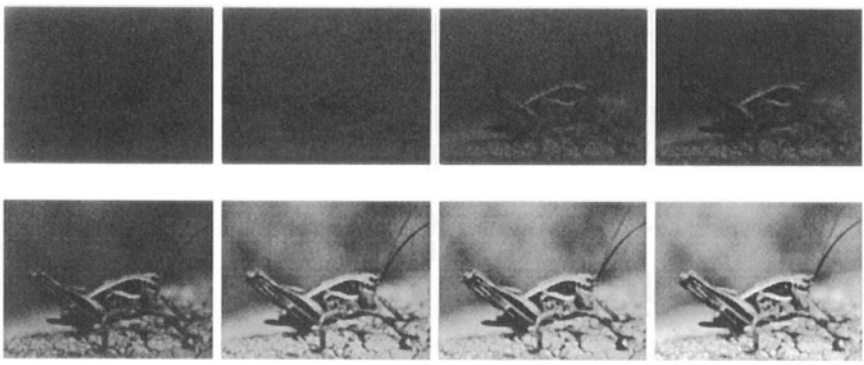

Figure 2-2. An example of the fade-in effect

2.2 SHOT-BOUNDARY DETECTION: UNRAVELING THE PROBLEM

2.2.1 Visual content discontinuities

The basic assumption when detecting shot boundaries in video is that the frames surrounding a boundary are, in general, much more different with respect to their visual content than the frames taken from within a shot. Note that a cut is surrounded by consecutive frames while a gradual transition is surrounded by the last unmodified frame from the first shot and the first unmodified frame of the second shot. For the reason of simplicity we will further assume that the modified frames in the case of a gradual transition do not belong to a shot but to the transition only. As we show in Figure 2-4, the frame n is considered the last frame of the shot i while the frame $n+1$ serves as the first frame of the shot $i+1$.

A high coherence of the visual content along the frames belonging to a shot is expected due to a high frame rate of digital video, which is typically in the range of 25 to 30 frames a second. Two consecutive frames of a shot are therefore likely to contain a considerable portion of the same visual material. This coherence is likely to be disturbed at the end of the shot as, after the shot boundary, the camera typically switches to another scene that is visually different from the scene in the frames preceding the boundary [Rei68].

Based on the above, the problem of detecting shot boundaries may generally be approached by searching for large discontinuities in the visual-content flow of a video. To do this, a *discontinuity value* $z(k)$ needs to be computed at each frame k to quantify the temporal variations of suitable

features of the visual content at the time stamp of that frame. The features are selected to depict those aspects of the visual content of a video that are most expressive regarding the appearances of shot boundaries, but that are, at the same time, rather insensitive to unimportant variations of the visual content along a shot. In the case of a cut, feature variation at the frame k can best be computed in relation to the next following frame $k+1$, as here an abrupt change in the visual content (and in the corresponding feature values) can be expected. Consequently, compared to relatively low values of $z(k)$ measured for the pairs of consecutive frames within a shot, much higher values of $z(k)$ may be expected when measuring feature variations between the frames surrounding a cut.

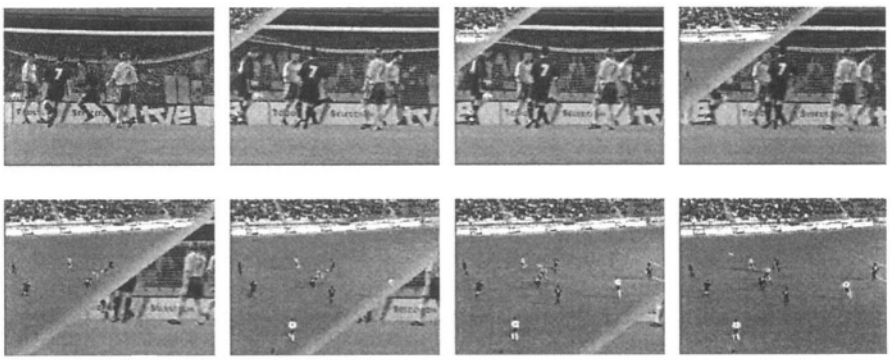

Figure 2-3. An illustration of the wipe effect

In the case of a gradual transition, the editing effect applied to the transition frames makes these frames visually different compared to the frames from both before and after the transition. For instance, the frames belonging to the wipe effect illustrated in Figure 2-3 differ from the frames surrounding the effect because they contain the portions of the visual material of both shots. Consequently, the $z(k)$ values that are computed for the pairs of consecutive frames surrounding the beginning and end time stamp of the editing effect, will likely be higher than those computed within a shot. Further, since the rate of the visual content change during the transition is typically higher than the visual content variations within a shot, higher discontinuity values are also expected when computed for the pairs of consecutive frames within the transition. As a result, not one distinguishable high discontinuity value is obtained like in the case of a cut, but a series of increased discontinuity values in the duration of the gradual transition.

As the rate of the visual content change across the transition is only gradual, the $z(k)$ values computed there are typically much lower than those

computed at cuts. In some cases, they may even not be distinguishable enough from the discontinuity values measured within a shot. In order to amplify the value range difference, an option is to compute the discontinuity values not for the consecutive frames but for the frames with a larger distance in-between. We will further refer to this distance as *inter-frame skip.* Figure 2-4 illustrates the computation of the discontinuity values for the inter-frame skip being equal to the length of the transition.

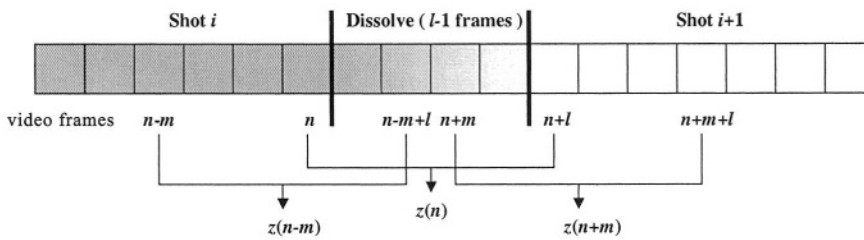

Figure 2-4. A gradual transition between two shots and the procedure for computing the discontinuity values with the inter-frame skip corresponding to the length of the transition

Clearly, obtaining a sufficient value range difference between the discontinuity values $z(k)$ computed within shots and at shot boundaries is the major prerequisite for being able to detect the presence of shot boundaries on the basis of these values, while avoiding the detection errors, that is, *missed* or *falsely detected* boundaries. In the following we will refer to these two value ranges as $\overline{R}$ and R, respectively. There are, however, many factors that can disturb the separation of discontinuity value ranges $\overline{R}$ and R. On the one hand, some special effects (e.g. morphing), screenplay effects (e.g. a person enters a dark room and turns on the light) and some extreme actions of a director (e.g. making extremely fast camera movements, pointing with the camera to a strong light source or to an object moving just in front of the camera) may cause the discontinuity values measured within a shot to be in the range that is typical for shot boundaries. An illustration of such an effect caused by extreme motion captured by the camera is shown in Figure 2-5. On the other hand, some discontinuity values measured at shot boundaries may be considerably lower than their typical value range. This may be the case, for instance, if the visual content of the shots attached to each other is too similar or if the rate of the content change during a gradual transition is insufficient.

2.2.2 Discriminative and prior information

In view of the discussion in the previous section we may conclude that, generally, reaching the optimal shot-boundary detection performance is not possible by only relying on the ranges of discontinuity values. We therefore introduce two types of additional information that can be used to compensate for the influence of the disturbing factors mentioned before: *discriminative information* and *prior information.*

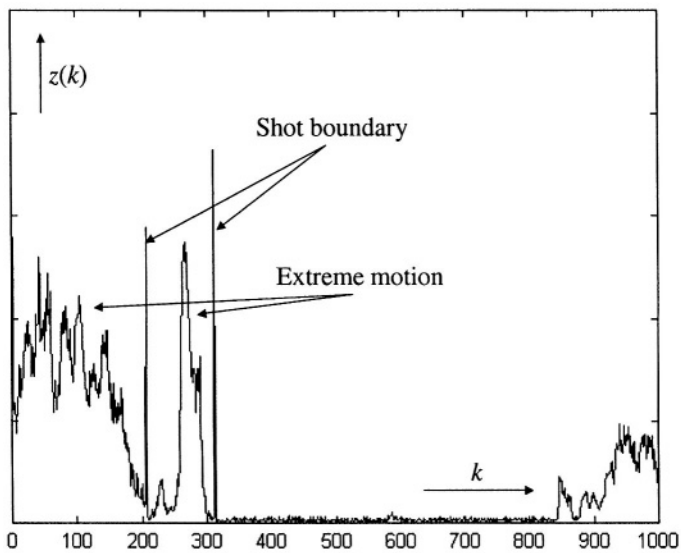

Figure 2-5. An illustration of problems regarding value range separation

The discriminative information can help distinguish the temporal behavior of discontinuity values at or around a particular shot boundary from the behavior of these values within a shot. In other words, one could match the temporal pattern created by a number of consecutive discontinuity values with the temporal pattern that is considered typical for a particular class of shot boundaries in order to check for the presence of a boundary from this class at a given time stamp. For instance, the appearance of an extremely high narrow peak in a series of discontinuity values is a strong indication for the presence of a cut at the place of the peak. Such peaks can be seen in the illustration in Figure 2-5. The expected improvement of the detection performance using this type of discriminative information is clearly a result of extending the "absolute" knowledge on a discontinuity value (its range)

by the "relative" knowledge on this value with respect to its local neighborhood. We refer to this class of discriminative information as *structural information.*

In our search for other forms of discriminative information we can also extend the consideration of the features beyond the sole objective of computing the discontinuity values. It is namely so that some features show typical behavior when measured in the frames around or within a shot boundary. For example, since a dissolve is the result of mixing the visual material from two neighboring shots, it can be expected that the values of intensity variance measured in the frames of a dissolve ideally reveal a downwards-parabolic pattern [Ala93]. We can now match this dissolve-related temporal behavior of the intensity variance with the temporal pattern generated by the variance values computed across a series of frames in order to check for the presence of a dissolve. This matching can be done either on the basis of characteristic pattern parameters, like for instance, *w* and *h,* as illustrated in Figure 2-6, or by using a mathematical model of the parabolic variance curve that is then fitted to the sequence of computed variance values. We refer to this class of discriminative information, which is drawn either from the parameters or a mathematical model of the temporal feature behavior characteristic for a particular shot boundary, as *feature information.*

As opposed to the discriminative information, the prior information indicating the presence or absence of a shot boundary at a certain time stamp along a video is not based on any direct measurement performed on a video, but rather on the general knowledge about the temporal structure of the video. For instance, we could intuitively assume that the probability of a new shot boundary immediately after the last detected boundary is negligible, but also that it increases with every further frame.

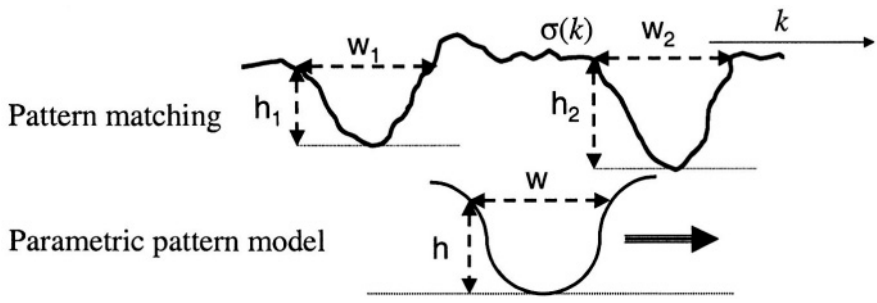

Figure 2-6. An example method for matching the intensity-variance pattern

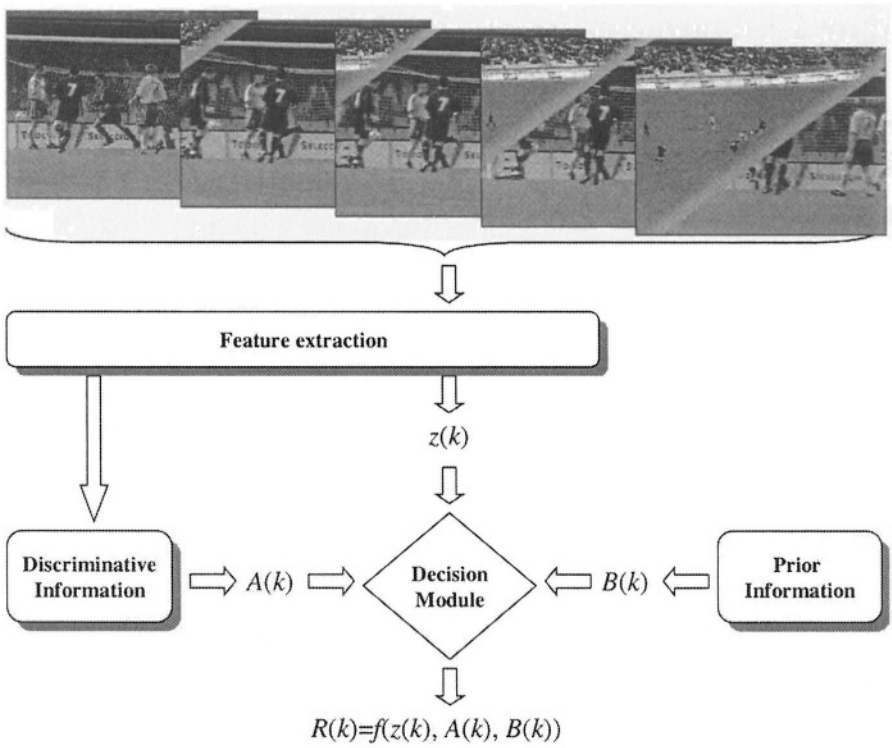

Figure 2-7. General scheme of a shot-boundary detector

2.2.3 Detector structure

After we identified all issues that are to be considered in the process of shot-boundary detection, the remaining challenge is to integrate these issues into a unified detection framework. A general scheme of such a framework is illustrated in Figure 2-7. The core of the framework is the decision module where it is decided about the presence or absence of a shot boundary at a given frame k on the basis of three uncorrelated inputs:

- Discontinuity values $z(k)$,
- Input $A(k)$ based on discriminative information,
- Input $B(k)$ based on prior information.

Generally, the inputs into the decision module originating from the discriminative and prior information can be considered time-dependent, as represented by the functions $A(k)$ and $B(k)$, respectively. This implies that

the decision mechanism is also time dependent, resulting in the decision function $R(k)$. The decision module processes the abovementioned three inputs and generates the values of the decision function at any given frame k. In Sections 2.3-2.6 we will discuss each of the four major blocks of the detector scheme in more detail.

2.3 FEATURE EXTRACTION

The success of shot-boundary detection largely depends on the selection of features used to represent the visual content flow along a video. As shown in Figure 2-7, features can be used to model either the discriminative information or the discontinuities in the visual content flow. In the first case, a feature needs to show a temporal behavior in or around a shot boundary, which is clearly distinguishable from its behavior within a shot. The intensity variance mentioned in Section 2.2.2 is a good example of such a feature. A feature that is to be used to compute the discontinuity values $z(k)$ needs to be evaluated with respect to its capability to secure a sufficient separation of discontinuity value ranges $\overline{R}$ and R.

We will discuss in this section the major classes of features regarding their suitability to serve for the two purposes mentioned above. We base our discussion on the surveys on feature extraction for shot-boundary detection presented in [Aha96], [Bim99], [Fur95] and [Lie99]. Further surveys can be found in [Idr97], [Man99] and [Lup98].

2.3.1 Pixel intensity

We will assume in general that the discontinuity value $z(k)$ is computed for the frames k and $k+l$, with $l \geq 1$. As first proposed by Kikukawa and Kawafuchi in [Kik92], the simplest way of measuring visual discontinuity between two frames is to compute the mean absolute intensity change for all pixels of a frame. If we denote the intensity of the pixel at coordinates (x,y) by $I(x,y)$ then the absolute intensity change of that pixel between frames k and $k+l$ is obtained as

$$D_{k,k+l}(x,y) = |I_k(x,y) - I_{k+l}(x,y)| \tag{2.1}$$

The discontinuity value $z(k)$ is then easily computed as the value (2.1) averaged over all frame pixels, that is, over the frame dimensions X and Y:

$$z(k) = \frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} D_{k,k+l}(x,y) \tag{2.2}$$

While the above feature is rather simple, it is also highly sensitive to camera and object motion. To better control the motion sensitivity, Otsuji et al. [Ots91] suggested to only count the pixels that change considerably from one frame to another. To do this, the absolute change of the intensity $I(x,y)$ is first compared with the prespecified threshold T, and is taken into account by the averaging formula (2.2) only if the measured absolute difference exceeds the threshold, that is

$$D_{k,k+l}(x,y) = \begin{cases} 1 & if\ |I_k(x,y) - I_{k+l}(x,y)| > T \\ 0 & else \end{cases} \tag{2.3}$$

A further reduction of the motion influence on the discontinuity values (2.2) can be obtained by following the proposal of Zhang et al. [Zha93], that is, to apply a 3x3 averaging filter to the frames before performing their pixel-wise comparison.

The best way of eliminating the influence of motion on the discontinuity values (2.2) is to perform motion compensation between the frames first, so that intensity changes are computed for the corresponding pixels. We will discuss this option in more detail in Section 2.3.5.

Instead of computing pixel-to-pixel intensity differences between two frames, we could also investigate how the statistics of pixel intensities change from one frame to another. An approach that can be used for this purpose is called the *likelihood ratio* approach [Kas91, Set95]. There, each of the two frames being compared is divided in regions and then the discontinuity value between the frames is computed on the basis of changes in the statistical properties (mean and variance) of the intensities per region. As here the global region characteristics are compared rather than single pixels, the robustness of the likelihood ratio approach regarding the noise induced by motion or illumination changes is somewhat larger than in the methods discussed before. A potential problem with this approach is that two different regions may still be similar in terms of their general pixel statistics.

When analyzing the changes in the statistics of pixel intensities from one frame to another, in parallel with searching for discontinuities we may also search for the temporal patterns that are typical for this statistics at places of shot boundaries. And, indeed, the variance of pixel intensities in a video frame shows a typical downwards-parabolic pattern (Figure 2-6) at places of dissolves [Ala93, Men95], and the upwards and downwards segments of this pattern at places of a fade-in and fade-out, respectively [Lie99]. This makes the pixel intensity variance a suitable feature for generating additional (discriminative) information for the shot-boundary detector. How this information can be modeled to provide input in the decision module, will be explained in Section 2.5.

2.3.2 Histogram

Another example of a feature that is based on pixel statistics, and that is frequently used for shot-boundary detection, is a histogram. Consecutive frames within a shot containing similar visual material will show little difference in their histograms, compared to the frames surrounding a shot boundary. Although it can be argued that frames having completely different visual contents can still have similar histograms, the probability of such a case is smaller than in the case where general statistical properties (mean and variance) are used. Since histograms ignore spatial changes within a frame, histogram differences are considerably more insensitive to object motion with a constant background than pixel-wise comparisons are. However, a histogram difference remains sensitive to camera motion, such as panning, tilting or zooming, due to large portions of new visual content introduced in each following video frame.

Histograms are used as features mainly for detecting the discontinuities in the visual content flow. This is done by bin-wise computing the difference between the histograms of neighboring video frames. Both grey-level and color histograms can be used, and their differences can be computed by a number of different metrics. A simple metric is the sum of absolute differences of corresponding bins. With N_{bins} being the total number of bins, this can be written as

$$z(k) = \sum_{j=1}^{N_{Bins}} | H_k(j) - H_{k+l}(j) | \tag{2.4}$$

when comparing grey-level histograms, and

$$z(k) = \sum_{C} \sum_{j=1}^{N_{Bins}} | H_k^C(j) - H_{k+l}^C(j) | \tag{2.5}$$

if color histograms are used [Yeo95b]. In (2.4), $H_k(j)$ is the j-th bin of the grey-value histogram belonging to the frame k. In (2.5), $H_k^C(j)$ is the j-th bin of the histogram of the C - component of the color space used.

Another popular metric for histogram comparison is the so-called χ^2-test [Nag92] that can generally be formulated as

$$z(k) = \sum_{j=1}^{N_{Bins}} \frac{| H_k(j) - H_{k+l}(j) |^2}{H_{k+l}(j)} \tag{2.6}$$

Zhang et al. [Zha93] reported that the metric (2.6) does not only enhance the discontinuities across a shot boundary but also the effects caused by camera/object motion. Therefore, the overall detection performance of (2.6) is not necessarily better than that from (2.4), whereas it does require more computational power.

We mention here also the histogram intersection as a further example of a frequently used metric for histogram similarity computation. The discontinuity value $z(k)$ can be defined on the basis of this intersection as

$$z(k) = 1 - \frac{\sum_{j=1}^{N_{Bins}} \min\left(H_k(j), H_{k+l}(j)\right)}{XY} \tag{2.7}$$

A discontinuity in the visual content flow can also be computed as the difference between the average colors of the histograms of neighboring video frames. The average color of a histogram H with bins j can be defined as a vector with the components

$$H_{avg}(C) = \sum_{j=1}^{N_{Bins}} H(j)\, c(C, j) \tag{2.8}$$

Here, C is again a component of the color space used, while $c(C, j)$ is the value of the component C at the histogram bin j [Haf95].

Gargi et al. [Gar00] have evaluated the cut-detection performance of the discontinuity values (2.5), (2.6), (2.7) and the one based on the average histogram color (2.8). The evaluation results show that the χ^2-test performs significantly worse than other metrics. Bad performance was also obtained for the discontinuities based on average color (2.8), while the histogram intersection formula performed best. The abovementioned metrics were computed in eight different color spaces, including RGB, HSV, YIQ, XYZ, L*a*b*, L*u*v*, the Munsell system of color notation, and the Opponent color space [Swa91, Fur95]. Although it was shown that the choice of the color space for histogram computation has less impact on the detection performance than the choice of a metric, the tests indicated that the Munsell color space was the best one, followed by the uniform color spaces L*a*b* and L*u*v* and the Opponent color space. Interestingly, all color spaces performed better than the luminance alone (grey-level histograms), which indicates a high importance of the color content of the video frame in detecting shot boundaries. While performing best, the Munsell color space is, however, also computationally most intensive. The L*a*b* color space

appears to be the best choice if the optimum between the detection performance and computational cost is searched for.

Histograms can also be computed and compared block-wise. In this case, both frames k and $k+l$ are divided into blocks, and the histograms $H_{k,i}$ and $H_{k+l,i}$ are computed for blocks $b_i(k)$ and $b_i(k+l)$. Then, the discontinuity value $z(k)$ can be found as a sum of block-wise histogram differences. Nagasaka and Tanaka [Nag92] divide a frame into 16 blocks and discard 8 largest differences to efficiently reduce the influence of motion and noise. An alternative to this approach can be found in [Ued91], where the number of blocks in the frame is increased to 48, and where – as opposed to [Nag92] - only those block-difference values are considered in the formula for $z(k)$ that exceed the prespecified threshold T, that is,

$$z(k) = \sum_{i=1}^{48} D\big(b_i(k), b_i(k+l)\big) \tag{2.9}$$

with

$$D\big(b_i(k), b_i(k+l)\big) = \begin{cases} 1 & if \ \dfrac{1}{N_{Bins}} \displaystyle\sum_{j=1}^{N_{Bins}} \frac{\big(H_{k,i}(j) - H_{k+l,i}(j)\big)^2}{H_{k,i}(j)} > T \\ 0 & else \end{cases} \tag{2.10}$$

Otsuji et al. [Ots93] found that the approach from [Ued91] is much more sensitive to abrupt boundaries than the one proposed in [Nag92]. However, since emphasis is put on blocks, which change most from one frame to another, the approach from [Ued91] also becomes highly sensitive to motion. To eliminate the influence of motion on the values $z(k)$, motion compensation can be applied to blocks before the method from [Ued91] is applied. This option will be discussed in more detail in Section 2.3.5.

2.3.3 Edges

Another characteristic feature class that proved to be useful in detecting shot boundaries is derived from edge statistics of a frame. Two edge-based features are frequently used in the context of shot-boundary detection:

- Edge-change ratio (ECR)
- Edge-based contrast (EC)

The edge-change ratio was first proposed by Zabih et al. [Zab95]. The underlying idea here is that, due to the change in the scene composition from

one shot to another, the edge set belonging to the objects found in the frames before the boundary will be different than the edge set of the objects in the new scene composition. To compute the ECR, first the overall motion between frames is computed. Based on the motion information, two frames are registered and the number and position of edges detected in both frames are compared. The total difference is then expressed as the total edge change percentage, i.e. the percentage of edges that enter and exit from one frame to another. Let p_k be the percentage of edge pixels in frame k, for which the distance to the closest edge pixel in frame $k+l$ is larger than the prespecified threshold T. In the same way, let p_{k+l} be the percentage of edge pixels in frame $k+l$, for which the distance to the closest edge pixel in frame k is also larger than the threshold T. Then, the discontinuity in the visual content flow based on the edge-change ratio is computed as

$$z(k) = \max(p_k, p_{k+l}) \tag{2.11}$$

An important advantage of the ECR feature is that it can successfully be applied in the detection of cuts, fades, dissolves and wipes [Zab95, Zab99]. The ECR value (2.11) exhibits namely typical patterns at places of shot boundaries. While cuts are clearly visible in the ECR time curve as sharp, highly distinguishable peaks, a series of high ECR values can be seen at places of gradual transitions. Thereby, the difference between fades on the one hand and dissolves and wipes on the other, is in the position of the local maximum of the obtained series of high ECR values. In the case a fade-in/fade-out this maximum is positioned at the beginning/end of the series. At dissolves or wipes, the local maximum can be found in the middle of the ECR value series [Lie99]. Another advantage of the ECR feature is that, due to the registration of frames prior to edge comparison, this feature is robust against motion. However, the complexity of computing the discontinuity values based on ECR is also high. In relation to this, the comparative studies presented in [Dai95], [Fer99b] and [Lie01b] indicate that the ECR feature, when employed for cut detection, does not outperform the histogram-based detection approaches.

The edge-based contrast feature was presented by Lienhart [Lie99] as an alternative tool for detecting dissolves. Due to the overlap of the visual material of two consecutive shots, the frames within a dissolve lose their contrast and sharpness compared to the frames surrounding a dissolve. Capturing and amplifying this loss is the basic idea behind the EC feature. Given the edge map $L(x,y,k)$ of the frame k, the threshold T_w for weak and the higher threshold T_s for strong edges, then the strengths of the strong and weak edge maps can be obtained using the following set of formulas:

$$w(k) = \sum_{x=1}^{X}\sum_{y=1}^{Y} W(x,y,k) \quad \text{and} \quad s(k) = \sum_{x=1}^{X}\sum_{y=1}^{Y} S(x,y,k) \tag{2.12}$$

with

$$W(x,y,k) = \begin{cases} L(x,y,k) & \text{if } T_w \le L(x,y,k) < T_s \\ 0 & \text{else} \end{cases} \tag{2.13a}$$

and

$$S(x,y,k) = \begin{cases} L(x,y,k) & \text{if } T_s \le L(x,y,k) \\ 0 & \text{else} \end{cases} \tag{2.13b}$$

Now, the EC value for the frame k is computed as

$$EC(k) = 1 + \frac{s(k) - w(k) - 1}{s(k) + w(k) + 1} \tag{2.14}$$

Because the underlying idea of the EC feature is similar to the one for the variance of pixel intensities, it is not surprising that a temporal pattern similar to the one in Figure 2-6 could be expected when computing the EC value in the frames belonging to a dissolve. In this sense, the EC value can be seen as an alternative to the intensity variance for generating the discriminative information for the shot-boundary detector.

2.3.4 Motion field

The features discussed in this section characterize the motion field that is measured between the neighboring video frames. As the continuity of the camera or object motion is inherent only in the frames within a shot, we can assume that a discontinuity in the visual content flow at an abrupt shot boundary is also characterized by a discontinuity in the motion field. An example of a feature belonging to this class is *motion smoothness* [Aku92]. Here, we first compute all motion vectors $\boldsymbol{v}^i_{k,k+1}$ between the frames k and $k+1$and then check their significance by comparing their magnitudes with a prespecified threshold T_1:

$$w_{i,1}(k) = \begin{cases} 1 & if \ |v^i_{k,k+1}| > T_1 \\ 0 & otherwise \end{cases} \tag{2.15a}$$

Then, we also take into consideration the frame $k+2$ and check whether a motion vector computed between frames k and $k+1$ significantly differs from the related motion vector measured between frames $k+1$ and $k+2$. This is done by comparing their absolute difference with a predefined threshold T_2:

$$w_{i,2}(k) = \begin{cases} 1 & if \ |v^i_{k,k+1} - v^i_{k+1,k+2}| > T_2 \\ 0 & otherwise \end{cases} \tag{2.15b}$$

The sum of the values (2.15a) over all motion vectors is the number of significant motion vectors between frames k and $k+1$, and can be understood as a measure for object/camera activity. Similarly, the sum of values (2.15b) is the number of motion vectors between frames k and $k+1$ that are significantly different from their corresponding vectors computed between the frames $k+1$ and $k+2$, and can be understood as the measure for motion discontinuity along three consecutive frames of a sequence. Using these two sums, we can now first compute the motion smoothness at frame k as

$$M(k) = \frac{\sum_{i=1}^{N_{Vectors}} w_{i,1}(k)}{\sum_{i=1}^{N_{Vectors}} w_{i,2}(k)} \tag{2.16}$$

and then the discontinuity value at frame k as the inverse of (2.16), that is

$$z(k,k+1) = \frac{1}{M(k)} = \frac{\sum_{i=1}^{N_{Vectors}} w_{i,2}(k)}{\sum_{i=1}^{N_{Vectors}} w_{i,1}(k)} \tag{2.17}$$

The features derived from the motion field of a video are among the most computationally expensive features discussed in the context of shot-boundary detection. However, as the motion vectors are available as side information in MPEG-compressed video data streams, the problem of increased complexity may be reduced if an MPEG-compressed video serves directly as input into the video analysis module.

2.3.5 Motion-compensated features

Regular object and camera motion, where no disturbing factors mentioned in Section 2.2.1 are involved, account for most of the variations in the visual content flow along a video. Examples of this type of motion are simple (and slow) camera panning, camera motion following a moving object and object motion by a stationary camera. Therefore, selecting the feature set that reduces or eliminates the motion influence on discontinuity values is likely to provide a significant contribution to obtaining sufficiently separated discontinuity value ranges $\overline{R}$ and R. Such feature set may also, however, contribute to a wide performance constancy of a shot-boundary detector. Namely, since different video genres can globally be characterized by specific average magnitudes of object/camera motion (e.g. high-action movies vs. stationary dramas), eliminating these distinguishing factors may provide a high level of *consistency* of ranges $\overline{R}$ and R across different genres. If the ranges $\overline{R}$ and R are consistent, the parameters of the detector can first be optimized on a set of training videos to maximize the detection reliability, and then the shot boundaries can be detected with the same high reliability in an arbitrary video without any human supervision.

The highest motion independence show the feature extraction approaches that are based on motion compensation. An example of such a technique was introduced earlier in the context of computing the edge change ratio (2.11). This ratio is computed after registering the frames, that is, after eliminating the changes in the edge structure due to camera motion. Another motion-compensated feature extraction method is based on a *block matching* procedure. This procedure is applied to find for each block $b_i(k)$ in frame k a corresponding block $b_{i,m}(k+l)$ in frame $k+l$, such that it is most similar to the block $b_i(k)$ according to a given criterion (difference formula) D, that is:

$$D_{k,k+l}(i) = D\left(b_i(k), b_{i,m}(k+l)\right) = \min_{j=1..N_{Candidates}} D\left(b_i(k), b_{i,j}(k+l)\right) \quad (2.18)$$

Here, $N_{Candidates}$ is the number of candidate blocks $b_{i,j}(k+l)$ considered in the procedure to find the best match for a block $b_i(k)$. If k and $k+l$ are neighboring frames of the same shot, then the values $D_{k,k+l}(i)$ can be assumed low. This is because for a block $b_i(k)$ almost the identical block $b_{i,m}(k+l)$ can be found due to the general constancy of the visual content within a shot. This is not the case if frames k and $k+l$ surround a shot boundary. Then, the difference between the "corresponding" blocks in these two frames will be large due to a substantial change in the visual content across the boundary. Thus, computing the discontinuity value $z(k,k+l)$ as a function of differences $D_{k,k+l}(i)$ is likely to provide a reliable base for detecting shot boundaries.

We illustrate the computation of the discontinuity values based on the results of the block-matching procedure by the technique proposed by Shahraray [Sha95]. There, a frame k is divided into $N_{Blocks} = 12$ non-overlapping blocks, and the differences $D(b_i(k), b_{i,j}(k+l))$ are computed by comparing pixel-intensity values between the blocks. Then, the obtained differences $D_{k,k+l}(i)$ are sorted and normalized between 0 and 1 (where 0 indicates a perfect match), giving the values $d^s_{k,k+l}(i)$ as results. These values are multiplied with weighting factors c_i to further refine the block-matching result and then combined together to give the discontinuity value:

$$z(k) = \sum_{i=1}^{N_{Blocks}} c_i d^s_{k,k+l}(i) \tag{2.19}$$

Clearly, the differences $D_{k,k+l}(i)$ between motion-compensated blocks $b_i(k)$ and $b_{i,m}(k+l)$ can also be computed using block histograms, for instance, by employing the expression (2.10).

2.4 MODELING PRIOR INFORMATION

The study of Salt [Sal73] on differences in styles of various film directors surprisingly showed that, with respect to the distributions of shot lengths, the diversity in styles for different filmmakers was not great. The obtained distributions were characterized in nearly all cases by a considerable similarity of their overall shapes. As reported by Salt, and later also confirmed by Coll and Choma [Col76] in the context of their analysis of "image activity characteristics in broadcast television", a large majority of obtained (normalized) distributions matched the Poisson model [Yat99]

$$P_X(x) = \frac{\mu^x}{x!} e^{-\mu}, \quad x = 0,1,2,... \tag{2.20}$$

Here, X represents the discrete "shot length" variable with the values corresponding to the number of frames in a shot. We denote by x an arbitrary value of the variable X, while μ is the model parameter. In probabilistic terms, X can be seen as a Poisson random variable and the expression (2.20) as its *probability mass function* (PMF).

The knowledge about the distribution of shot lengths in a video could serve as the source of prior information that could be used to enhance the shot-boundary detection performance. And, indeed, we can model the prior probability $P_k(S)$ of a shot boundary S occurring at the frame k on the basis of the distribution model (2.20).

Let $x(k)$ *be* the current shot length, measured at the frame k since the last detected shot boundary. Evaluating the PMF (2.20) at $x(k)$ gives us the probability that the length of the analyzed shot is equal to $x(k)$. As this probability, however, not always increases with increasing frame index k, the PMF (2.20) is not directly suitable as the model for $P_k(S)$. While, namely, being close to zero at the beginning of a shot, the probability $P_k(S)$ should increase monotonously with each further frame elapsed since the beginning of the shot, and converge toward 1 for $k \to \infty$. We, therefore, use the *cumulative distribution function* (CDF) of the random variable X as a model for the probability $P_k(S)$. First, in view of the required behavior and value range specified above, we may say that $P_k(S)$ is inherently a CDF. Second, as the CDF value for the argument $x(k)$ is defined as the probability that the random variable X is no larger than $x(k)$, this probability can be seen as equivalent to the probability that the analyzed shot will not get any longer than $x(k)$ or, in other words, that a shot boundary will occur at the observed frame k. With increasing frame index k and absence of a shot boundary, the CDF value for the argument $x(k)$ will rise toward one. In the limit case for $k \to \infty$ the shot has become longer than any shot observed before (e.g. in the training set) and, therefore, the probability that the shot will get any longer becomes close to zero.

A slight adjustment of the cumulative probability model is required, however, in order to limit the influence of the prior information on the detection performance in the case of an unusually long shot. Namely, a false boundary may be found in such a shot when the CDF arrives into the value range close to 1. To avoid this, the CDF can be scaled from its original value range [0, 1] onto the range [0, 0.5]. Then, in the limit case for $k \to \infty$ the influence of the prior information is negligible as the probability of a shot boundary becomes close to 50%.

Based on the above discussion, a model of the prior probability $P_k(S)$ of a shot boundary can now be defined as

$$P_k(S) = \frac{1}{2} \cdot \sum_{x=0}^{x(k)} P_X(x) \tag{2.21}$$

Clearly, the major role of the prior probability model (2.21) in the shot-boundary detection process is to prevent the detection of new shot boundaries shortly after the last detected boundary. Although the effectiveness of the model (2.21) based on the Poisson model for shot duration was demonstrated in [Han02], the reader is suggested to also consult the alternative ways of modeling prior knowledge, such as the approach based on the Weibull model for shot-length distribution proposed by Vasconcelos and Lippman [Vas00].

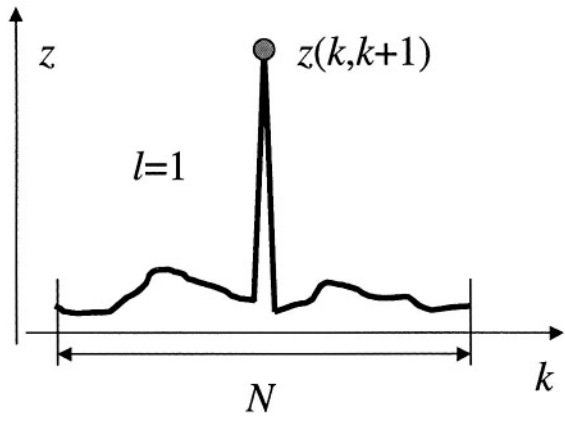

Figure 2-8. Expected behavior of N consecutive discontinuity values in the neighborhood of an abrupt shot boundary (cut) occurring between the frames k and $k+1$.

2.5 MODELING DISCRIMINATIVE INFORMATION

In addition to the prior knowledge regarding the presence of a shot boundary at a given time stamp in video, we may also use different types of *discriminative information* to enhance the shot-boundary detection performance. In Section 2.2.2 we defined two general classes of such information, namely the *structural* and *feature* information. In the following sections we show how discriminative information can be modeled by a *discriminative function* $\psi(k)$, the values of which indicate the occurrence of a shot boundary of a particular type at the frame k. Due to the fact that the function $\psi(k)$ will contain information that is characteristic for a particular boundary type, it needs to be computed for each boundary type individually. We discuss in this section the possibilities of defining the discriminative function $\psi(k)$ for both general classes of shot boundaries, namely the abrupt and the gradual ones.

2.5.1 Discriminative function for abrupt boundaries

As we explained in Section 2.2.2, the presence of a particular shot boundary type at the frame k may be revealed by the pattern created by a number of consecutive discontinuity values $z(k)$ computed in the neighborhood of the frame k. Yeo and Liu [Yeo95b] showed that, if the discontinuity values are computed between the consecutive frames of video, the presence of an isolated sharp peak surrounded by low discontinuity

values, like the one illustrated in Figure 2-8, may be seen as a reliable indication for the presence of a cut at the position of the peak. To effectively employ this information for shot-boundary detection, Yeo and Liu propose the procedure illustrated in Figure 2-9. There, N last computed discontinuity values are considered that form a *sliding window*. As we wish to check the presence of a cut at the frame k we denote the discontinuity value in the middle of the window by $z(k,k+1)$. The presence of a cut is checked for at each window position, in the middle of the window, according to the following criterion:

$$\text{if } \left\{ \begin{array}{l} z(k,k+1) = \max\limits_{i=-\frac{N}{2},\ldots,\frac{N}{2}} \left(\forall z(k+i,k+1+i)\right) \\ z(k,k+1) \geq \alpha z_{sm} \end{array} \right\} \Rightarrow \textit{cut between } k \textit{ and } k+1 \tag{2.22}$$

The criterion (2.22) specifies that a cut is detected between frames k and $k+1$ if the discontinuity value $z(k,k+1)$ is the window maximum and, at the same time, α times higher than the second largest discontinuity value z_{sm} within the window. The parameter α can be understood as the *shape parameter* of the pattern generated by the discontinuity values in the sliding window. Applying (2.22) at each position of the sliding window is, namely, nothing else but matching the ideal pattern shape with the actual behavior of discontinuity values found within the window. The major weakness of this approach is the heuristically chosen and fixed parameter α. Because α is fixed, the detection procedure is too coarse and too inflexible, and because it is chosen heuristically, one cannot make statements about the scope of its validity.

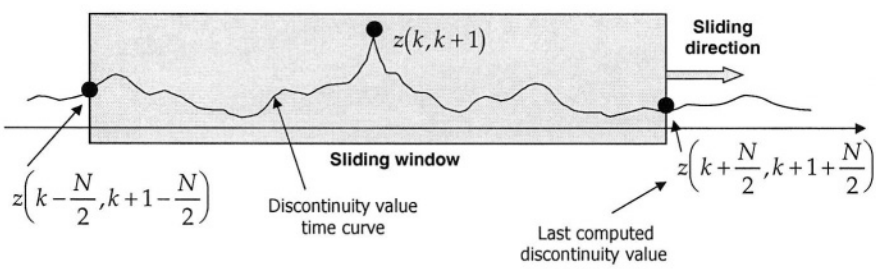

Figure 2-9. An illustration of the approach to shot-boundary detection from [Yeo95b]

Instead of coarsely comparing the ratio between $z(k,k+1)$ and z_{sm} with a fixed threshold we could interpret this ratio solely as an indication for the match between the measured and the template pattern. The larger the ratio, the better is the match, that is, the stronger is the indication of the cut presence. In view of this, the simple ratio between $z(k,k+1)$ and z_{sm} could already serve as a discriminative function $\psi(k)$ for cuts. However, as it is difficult to relate a particular value of this ratio to the strength of the indication for the presence of a shot boundary, the relative distance between $z(k,k+1)$ and z_{sm} appears better suitable for this purpose [Han02].

Based on the above, the discriminative function $\psi(k)$ for the detector of abrupt shot boundaries can be defined as

$$\psi(k) = \begin{cases} 100 \cdot \dfrac{z(k,k+1) - z_{sm}}{z(k,k+1)} \% & \text{if } z(k,k+1) = \max\limits_{i \in \left[-\frac{N}{2}, \frac{N}{2}\right]} \left(\forall z(k+i, k+i+1)\right) \\ 0 & \text{else} \end{cases} \tag{2.23}$$

With its values expressed in percentages, the function (2.23) evaluates the pattern matching in the middle of the sliding window of the length N and centered at the discontinuity value $z(k,k+1)$. Since the necessary condition for the presence of the cut between frames k and $k+1$ is that a sharp, isolated peak is found at $z(k,k+1)$, no boundary can be found there if $z(k,k+1)$ is not the maximum of the window. The length N of the window should be as large as possible in order to obtain reliable pattern matching results. However, in order to prevent the situation where the sliding window captures two shot boundaries, the maximum value of N should be limited by the minimum expected shot length.

2.5.2 Discriminative function for a gradual boundary

Compared to the case of abrupt shot boundaries, defining a reliable function that represents the structural discriminative information for the detector of gradual transitions is considerably more difficult. This is mainly due to the fact that the pattern of consecutive discontinuity values at a particular gradual transition is not unique and may vary in both shape and length. The variations of the pattern length are directly related to the varying length of the transition. The variations of pattern shape are mainly due to the effects occurring simultaneously with the transition, such as object and camera motion. The shape variations of gradual boundary patterns are the main reason for the difficulty to recognize such patterns using the methods that are based on pattern modeling (e.g. [Ala93, Aig94, Ham94, Kob99]).

A way of dealing with shape variations of boundary patterns when defining the structural component $\psi_1(k)$ of the discriminative function $\psi(k)$ would be to only capture in a model the global shape characteristics of the pattern. Although these characteristics may contain only a fraction of the entire information about the pattern and are, therefore, not discriminative enough, they could be sufficiently powerful to indicate the presence of a boundary candidate. Then, the final decision regarding the presence of a particular boundary type at the time stamp indicated by the candidate can be made on the basis of the feature component $\psi_2(k)$ of the discriminative function $\psi(k)$.

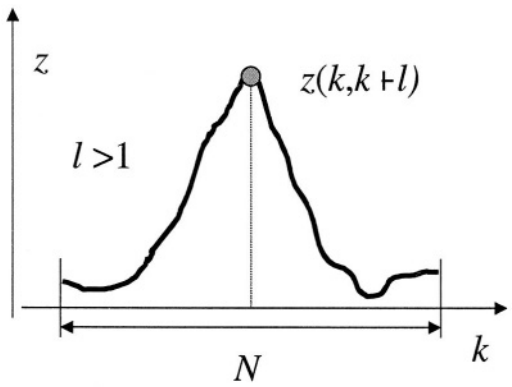

Figure 2-10. An illustration of the behavior of N consecutive discontinuity values computed for the inter-frame skip l>1 within a dissolve. A close-to-triangular pattern is expected if the dissolve length is close to the value of l.

We illustrate the possibilities for employing a coarse model of a boundary pattern for defining the function $\psi_1(k)$ on the following approach that addresses the problem of dissolve detection [Han02]. First, the discontinuity values are computed with the inter-frame skip l>1 using motion-compensated color differences (2.18) of the pixel blocks in the frames k and $k+l$. If the value of l is close to the dissolve length, the consecutive values $z(k,k+l)$ are expected to form a close-to-triangular shape as shown in Figure 2-10. Here, again the sliding window of N last computed discontinuity values is considered, centered at the value $z(k,k+l)$. Since the slopes of the triangular pattern are not pronounced, there is little sense in trying to model the pattern precisely and to measure the degree of pattern matching based on that model. Therefore the criterion for pattern matching addresses here only the global characteristics of the measured boundary pattern, and is defined as follows:

- The middle window value $z(k,k+l)$ is the maximum value of the window,

- Window's maxima on each side of $z(k,k+l)$ have to be sufficiently close to the middle of the window.

Clearly, we require that the pattern created by discontinuity values matches the "ideal" pattern of a dissolve only regarding the "top" of the triangular shape in Figure 2-10 and do not consider the shape of the slopes. Based on the above, the structural component of the discriminative information for dissolves can now be represented by the function $\psi_1(k)$:

$$\psi_1(k) = \begin{cases} 1, & if \;\; z(k,k+l) = \max\limits_{i \in \left[-\frac{N}{2},\frac{N}{2}\right]} (\forall z(i,i+l)) \;\; \wedge \;\; \Delta^l_{\max} + \Delta^r_{\max} < \frac{N}{2} \\ 0, & else \end{cases} \tag{2.24}$$

Here, $\Delta^l_{\max}$ and $\Delta^l_{\max}$ are the distances of the largest discontinuity values to the left and to the right of $z(k,k+l)$ from the window middle point. As can be seen from the condition in (2.24), the value of $\psi_1(k)$ is set equal to 1 at the frame k if the pattern created by discontinuity values of the sliding window fulfills the two matching criteria listed in the items above. Otherwise, $\psi_1(k)$ is set to 0. Consequently, each series of discontinuity values for which $\psi_1(k)$ is set equal to 1 is further considered as a dissolve candidate. One should note, however, that the discriminative quality of the function (2.24) is sensitive to the difference between the inter-frame skip l and the dissolve length. Due to the assumption about a close-to-triangular shape of the dissolve pattern, l should be selected as similar to the expected dissolve length. Generally, the function (2.24) is expected to react properly as long as the dissolve is not shorter than the half of the inter-frame skip l. If the dissolve is too short, then the triangular discontinuity pattern will be too flattened for the condition involving the distances $\Delta^l_{\max}$ and $\Delta^l_{\max}$ to work. On the other hand, if the dissolve is longer than the inter-frame skip, then the discontinuity values that are computed by comparing the frame pairs from within the transition may be smaller than those involving frames surrounding the beginning and the ending time stamp of the dissolve. In that case, the function (2.24) is not likely to work properly.

In order to check for the presence of a shot boundary at the time stamp indicated by the boundary candidate, the feature component $\psi_2(k)$ of the discriminative function can be used. The information considered by this component serves not only to compensate for the imperfections in the

discontinuity value patterns of a gradual transition but also to better distinguish that transition from other phenomena. It is namely so that the irregular discontinuity value patterns that are due to extreme factors can often be falsely classified as those belonging to gradual transitions because of their similarity with the noisy transition patterns (see the explanation in Section 2.2.1 and the example in Figure 2-5).

In previous sections, we have already discussed the potential suitability of the pixel-intensity variance as a source of discriminative information for supporting the shot-boundary detection. The variance-based approach for dissolve detection was first proposed by Alattar in [Ala93] but has been used and modified by other authors as well (e.g. [Fer99b, Gu97, Men95, Tru00a, Tru00b, Lie01a]). The detection of a dissolve is reduced here to detecting the parabolic curve pattern in a series of measured variance values. One simple example of the techniques that can be applied for this purpose was already explained in Figure 2-6. Realizing this in practice is, however, rather difficult as the two large negative spikes of the parabolic pattern are generally not sufficiently pronounced due to noise and motion in video. This problem was addressed by Truong et al. [Tru00a, Tru00b], who proposed an improved version of the variance-based detector of Alattar [Ala93]. Truong et al. proposed to exploit the facts that

- the first derivative of the pattern should be monotonically increasing from a negative to a positive value,
- the intensity variances of both shots involved should be larger than a given threshold,
- the dissolve duration usually falls between two well-defined thresholds.

Hanjalic [Han02], however, considered the variance pattern too imperfect for the approaches proposed by Alattar and Truong. Therefore, just like in the process of modeling the function $\psi_1(k)$, he chose to match only some of the characteristic global properties of the pattern. Namely, if the sliding window captures a dissolve, the variance measured for frames in the middle of the window will be considerably lower than the variance of frames positioned near window's edges. In contrast to this, if no dissolve occurs in a sliding window, the variance is expected to remain relatively stable across the window. On the basis of the above, Hanjalic computed the variance-based feature component $\psi_2(k)$ of the discriminative function as the relative change of intensity variance in the frame from the middle of the sliding window compared to the variance in the frames close to window edges.

$$\psi_2(k)=\begin{cases} 100\cdot\left(1-\dfrac{\min\left(\sigma(k),\sigma\left(k+\frac{N}{2}\right)\right)}{\max\left(\sigma(k),\sigma\left(k+\frac{N}{2}\right)\right)}\right)\%, & \text{if } \dfrac{\left|\left(\sigma(k)-\sigma\left(k+\frac{N}{2}\right)\right)\right|}{\left|\left(\sigma(k)-\sigma\left(k-\frac{N}{2}\right)\right)\right|}\le 1 \\ 100\cdot\left(1-\dfrac{\min\left(\sigma(k),\sigma\left(k-\frac{N}{2}\right)\right)}{\max\left(\sigma(k),\sigma\left(k-\frac{N}{2}\right)\right)}\right)\%, & \text{if } \dfrac{\left|\left(\sigma(k)-\sigma\left(k-\frac{N}{2}\right)\right)\right|}{\left|\left(\sigma(k)-\sigma\left(k+\frac{N}{2}\right)\right)\right|}<1 \\ 0 & \text{if } \dfrac{\sigma(k)}{\sigma\left(k-\frac{N}{2}\right)}>1 \text{ and } \dfrac{\sigma(k)}{\sigma\left(k+\frac{N}{2}\right)}>1 \end{cases} \tag{2.25}$$

Equation (2.25) formulates $\psi_2(k)$ analytically. Here, $\sigma(k)$ is the variance of the frame in the middle of the window with, while $\sigma(k\text{-}N/2)$ and $\sigma(k+N/2)$ are the variances of the frames at both window edges. The third option in the equation (2.25), where $\psi_2(k)$ is by definition equal to 0, corresponds to the case where variance in the middle of the window is larger than the variances at window edges. Since this cannot occur in a downwards-parabolic pattern, such relation among three variances cannot reveal a dissolve.

The values of the function $\psi_2(k)$ can therefore be seen as a reliable indication for the presence of a dissolve in a candidate series of discontinuity values selected using function $\psi_1(k)$. Merging the structural and feature components of the discriminative information into a unified discriminative function $\psi(k)$ for dissolve detection can now simply be done by multiplying the functions $\psi_1(k)$ and $\psi_2(k)$, that is

$$\psi(k)=\psi_1(k)\cdot\psi_2(k) \tag{2.26}$$

As discussed in Section 2.3.3, edge-based features can serve as an alternative to the variance of pixel intensities for deriving discriminative information about the presence of a shot boundary at a given time stamp. As explained by Lienhart [Lie01b], the edge contrast (EC) shows similar behavior during a dissolve as the variance of pixel intensities. Further, the ratio between the values of the edge-change ratio (ECR) measured at the beginning and end of a gradual transition can serve as a feature component of the discriminative function for detecting fades, dissolves and wipes [Zab95, Zab99].

2.5.3 Probabilistic embedding of discriminative information

As the main purpose of discriminative information is to provide an indication regarding the presence of a particular shot boundary at a given time stamp, this indication can, just like in the case of prior information, most naturally be formulated using probabilistic terms. A convenient way of doing this would be to define the conditional probability $P(S_b|\psi_b(k))$ of a shot boundary of the type b at the frame k given the value of the corresponding discriminative function $\psi_b(k)$ computed in the neighborhood of the frame k.

In order to properly embed the relation between the values of $\psi_b(k)$ and the probability of a shot boundary S_b, the function $P(S_b|\psi_b(k))$ needs to satisfy three major requirements. First, as a high value of $\psi_b(k)$ should lead to a high probability of boundary occurrence, $P(S_b|\psi_b(k))$ is clearly a monotonously increasing function of $\psi_b(k)$. Further, being a probability, $P(S_b|\psi_b(k))$ needs to be defined such that its value range covers the interval [0,1]. Finally, in order to be sufficiently robust, the function $P(S_b|\psi_b(k))$ should not be too sensitive to the values of $\psi_b(k)$ being close to boarders of the interval $[0, \psi_{max}]$. While for $\psi_b(k)$ being close to 0 or ψ_{max} the probability $P(S_b|\psi_b(k))$ should be almost 0 or 1, respectively, the actual transition from 0 to 1 should take place in the middle range of the interval $[0, \psi_{max}]$, that is, for values of $\psi_b(k)$ for which the boundary characteristics become distinguishable. This transition should, however, not be abrupt but flexible enough in order not to reject any reasonable boundary candidate.

Clearly, we could define the discriminative function $\psi_b(k)$ in the way that it already satisfies all of the above criteria, like, for instance, the functions (2.23) and (2.26) introduced as examples in the previous section. In that case, the function $\psi_b(k)$ can directly be used as the probability $P(S_b|\psi_b(k))$. However, even if $\psi_b(k)$ cannot be defined in this way, a variety of suitable functions can be used taking the values of $\psi_b(k)$ as arguments. As an example, we present here the function $P(S_b|\psi_b(k))$ formulated analytically as follows [Han02]:

$$P(S_b \mid \psi_b(k)) = \frac{1}{2}\left(1 + erf\left(\frac{\psi_b(k) - d}{\sigma_{erf}}\right)\right) \tag{2.27}$$

with

$$erf(x) = \frac{2}{\pi}\int_0^x e^{-t^2} dt \tag{2.28}$$

An illustration of the function (2.27) can be found in Figure 2-11. The parameters d and σ_{erf} are the "delay" from the origin and the spread factor determining the steepness of the middle curve segment, respectively. The optimal parameter combination (d, σ_{erf}) can be found experimentally, for instance, such that the detection performance on the training data set is optimized for the selected parameter values. In general, since $\psi_b(k)$ is computed differently for each boundary type, the parameter combination (d, σ_{erf}) would also need to be determined for each boundary type separately. The basic shape of the conditional probability $P(S_b|\psi_b(k))$, however, can be considered the same for all boundary types.

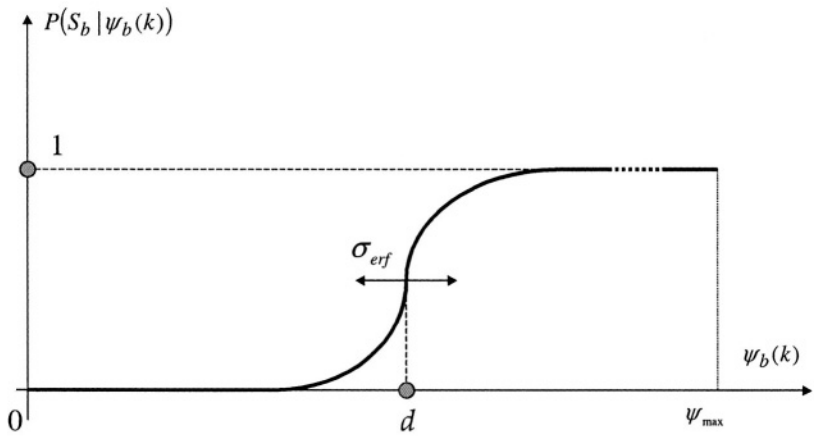

Figure 2-11. A probabilistic model for discriminative information [Han02]

2.6 BAYESIAN APPROACH TO DECISION MODULE DESIGN

In the broadest sense, the shot-boundary detection can be seen as a pattern classification problem that assigns a given temporal segment of a video either to the category "no boundary" (clearly, if no boundary is contained in that segment) or to a category corresponding to a specific boundary type (like "cut", "dissolve", "wipe", or "fade"). In view of the approaches discussed in Sections 2.5.1 and 2.5.2, the temporal video segment mentioned above can be seen as the segment captured by the sliding window at a given time stamp. Consequently, a wide range of tools known from the area of pattern classification can be used to design a shot-boundary detector.

In view of the availability of the prior and discriminative knowledge, the influence of which on the detection process is most naturally formulated using probabilistic terms, we can say that the problem of shot-boundary detection is inherently a probabilistic problem. Therefore, the *Bayesian decision theory* [Dud01] appears to be the natural framework for developing the shot-boundary detector, or, in our specific case, the decision module in Figure 2-7.

In the general application context of video content analysis, both types of detection errors, that is, *false* and *missed detections,* may have equal negative influence on subsequent high-level video content analysis steps. In this sense, the performance quality of the detector is determined by the average probability that any of the errors occurs. We therefore choose the minimization of the average error probability as the criterion to illustrate the development of a Bayesian shot-boundary detector.

We now consider the temporal video segment captured by the sliding window at the time stamp *k,* and search for the best category for that segment, given the discontinuity values within the window, the prior and the discriminative information. As can be derived from the general Bayesian decision rule, minimizing the average error probability corresponds to deciding for the category c^k_{opt} that maximizes the posterior probability $P_k(c_i|z(k))$ of the category c_i, that is

$$c^k_{opt} = \arg\max P_k(c_i \mid z(k)) \tag{2.29}$$

where $z(k)$ is the discontinuity value measured at frame *k,* and where c_i may represent any of the hypotheses of the posed shot-boundary detection problem, such as “cut”, “dissolve”, “wipe” or “no boundary”.

By applying the Bayes rule to the posterior probability $P_k(c_i|z(k))$ the decision rule (2.29) can be written as

$$c^k_{opt} = \arg\max\{P_k(c_i)P(z(k) \mid c_i)\} \tag{2.30}$$

Here, $P_k(z(k)|c_i)$ is the *likelihood* of the category c_i with respect to the observed discontinuity value, while $P_k(c_i)$ is the prior probability that the category c_i is the proper one to be assigned to the observed video segment.

The likelihood $P_k(z(k)|c_i)$ can be estimated through the following two steps:

- Generating normalized distributions of the discontinuity values for each category on the basis of training video data,

- Finding the parametric models approximating the obtained distributions.

In their technique developed for abrupt boundary detection, Vasconcelos and Lippman [Vas00] approached the second step using *generic mixture models* [Dud01]. They found a mixture of Erlang distributions an appropriate model for the "no boundary" category, and a combination of a Gaussian and a uniform density as the best model for the "cut" category. A mixture of two Gaussian distributions was used by Boreczky and Wilcox [Bor98] as the basis for modeling the likelihood for a fade, while all other categories were represented by simple models involving one Gaussian density only. Hanjalic [Han02] also proposes simple one-density models of the "cut", "dissolve" and "no boundary" categories.

By investigating the classification rule (2.30) more closely, one can realize that this rule does not explicitly take into account the discriminative information that we modeled in Section 2.5. A way of integrating this information in the detection process is to see the prior probability $P_k(c_b)$ of the category c_b representing the shot boundary of the type b as the joint probability of two events occurring at the frame k, namely the event that any shot boundary occurs at the frame k, and the event that this boundary is of the type b. As the occurrence of the latter event only depends on the value of the discriminative function $\psi_b(k)$, the two events can be considered independent. Using the results from sections 2.4 and 2.5 we can now define the new prior probability of a boundary of the type b at the frame k as

$$P_k(c_b) \equiv P_k(S, c_b) = P_k(S) \cdot P(c_b \mid \psi_b(k)) \tag{2.31}$$

Clearly, the conditional probability $P(c_b|\psi_b(k))$ can be seen as a modifier of the general prior probability $P_k(S)$ of shot boundary occurrence in view of the "context" defined by the boundary class b. In this sense, the expression (2.31) can be referred to as *context-dependent prior probability* of shot boundary occurrence. The effect of the discriminative information on the shot-boundary detection process becomes apparent in the situations where both $P_k(S)$ and the likelihood $P_k(z(k)|c_b)$ are in favor of signaling the presence of the boundary type c_b (e.g. when a large likelihood value appears long after the last detected boundary), whereby c_b is not the proper category to select at the given time stamp. In this way, the boundaries detected falsely due to, for instance, disturbing factors mentioned in Section 2.2.1 can be prevented using the discriminative information that is embedded in $\psi_b(k)$.

Since the prior probabilities of all categories add up to one, we can now express the prior probability of the remaining category "no boundary" as

$$P_k(c_j = \text{"no boundary"}) = 1 - \sum_{b \neq j} P_k(c_b) \tag{2.32}$$

An alternative to the above concept of multi-hypotheses testing is to use cascaded *binary detectors* [Han02], as illustrated in Figure 2-12. A binary detector decides about the presence of the boundary of one particular type only. If that boundary type is not found at the given time stamp, then the detector of another boundary type is activated.

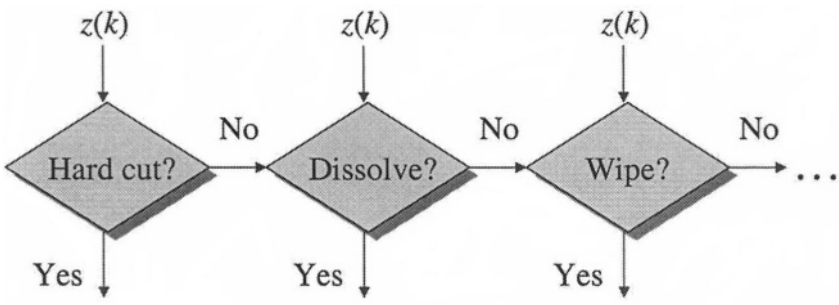

Figure 2-12. Cascaded binary detectors [Han02]

Except of keeping the basic detector structure simpler than in the general case (2.29), linking the detectors as described above can also be beneficial for improving the total detection performance of the cascade, especially if different inter-frame skips are used per detector. We explain this on the example of two series of discontinuity values computed for the inter-frame skip l=1 and l=22, that are aligned in time as shown in Figure 2-13.

In addition to its usefulness for amplifying the changes in the visual content flow along a gradual transition, a large value of the inter-frame skip produces "plateaus" that surround the high peaks of the cuts. Due to the appearance of the series of large discontinuity values, each of these plateaus can be mistaken for a gradual transition, that is, a falsely detected gradual transition can be reported at a certain plateau point, before or after a cut is detected. However, if the construction in Figure 2-12 is used, and if the detector of cuts is the first one in the cascade, the cuts are detected first. Then, all gradual transitions detected in later cascade components and found within the interval $(k\text{-}l/2, k+l/2)$ from a cut can be assumed a consequence of a plateau and, therefore, eliminated a posteriori. Hereby, the probability to eliminate a valid gradual transition is almost negligible since, first, the

detection of cuts is, in general, rather reliable and, second, the occurrence of a cut this close to a gradual transition is highly improbable.

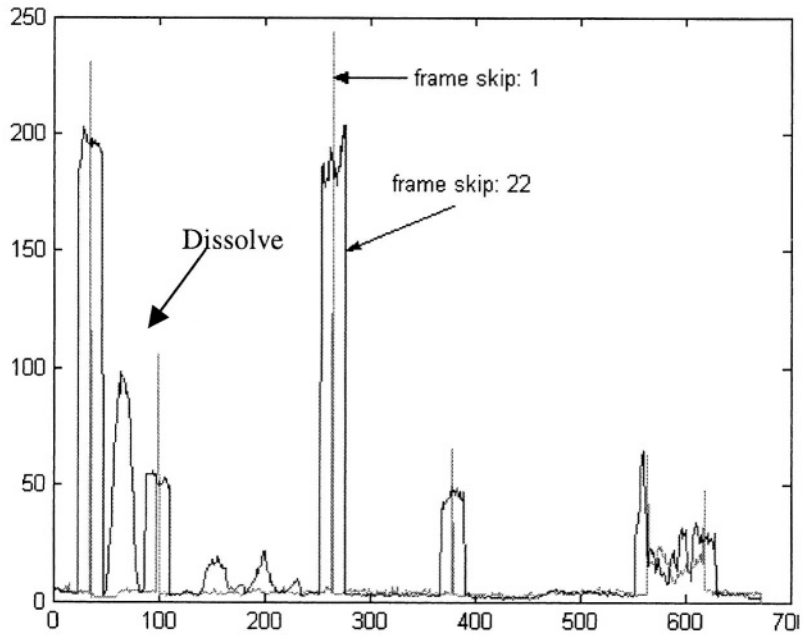

Figure 2-13. Discontinuity value time curves obtained on the basis of motion-compensated pixel block differences with the inter-frame skip l=1 and l=22 [Han02].

2.7 REMARKS AND RECOMMENDATIONS

Although the theory and algorithms for shot-boundary detection have reached a high level of maturity, the problem of shot-boundary detection is still not completely solved. The challenges can be summarized as follows:

- Understanding gradual transitions: What are the properties that make a particular class of gradual transitions considerably different from other transition classes and other effects in video?

- How to model the characteristic properties of a transition in the form of discriminative information that can serve as a constructive input into a shot-boundary detector?

- What can we do more regarding the feature selection and computation of discontinuity values in order to make the shot-boundary detection more robust toward special effects, screenplay effects and extreme actions of the director?

- How to minimize the complexity of the shot-boundary detection down to the level where (close-to) real-time implementation of the detector becomes possible, while keeping the detection error rate sufficiently low?

Let us now discuss these challenges in more detail:

While detecting cuts is not that much of a problem any more, the available detectors for gradual transitions are still of rather insufficient quality. This is because of great diversity of the measurable signal behavior around and within these transitions. The diversity stems in part from varying video-directing styles, and in part from multiple superimposed effects, like in the case of a gradual boundary accompanied by a lot of object or camera motion. In this chapter we showed how the discriminative information can be derived in the case of a dissolve. The discriminative quality of the function modeling this information is, however, dependent on the dissolve length. Besides, the discriminative functions for other types of gradual transitions are still to be defined. This needs not to be done from scratch. A long history of previous attempts to detect these transitions can certainly serve as a valuable source of inspiration.

In Section 2.6 a Bayesian minimum-error-probability (MEP) criterion was used as the basis for developing the shot-boundary detector. The main motivation to develop this detector within the framework of Bayesian decision theory was the need to embed the prior and discriminative information that are both expressed most naturally in probabilistic terms. The MEP criterion was chosen to maximize the detection performance with respect to the number of detection errors, that is, false and missed boundaries, which could have negative effect on subsequent higher-level video content analysis steps. However, alternatives have been proposed as well. For instance, Boreczky and Wilcox [Bor98] consider another class of Bayesian techniques – hidden Markov models - to design their shot boundary detector, while Lu and Zhang [Lu99b] employ neural network tools for this purpose. Clearly, the entire range of pattern classification techniques can be employed when designing a shot-boundary detector, as long as ways are found to effectively embed the prior and discriminative information in the detection process.

Last but not least, we address the issue of computational complexity of shot-boundary detection algorithms. In video content analysis systems shot-boundary detection is generally considered only a preprocessing step that provides the information on the basic temporal video structure to be used in much more complex, high-level content analysis algorithms. As these algorithms take most of the available computational power, the complexity of the preprocessing steps should be kept low. However, minimizing the complexity is, generally, a contradictory requirement to the one of securing a low detection-error rate. A good example is the abovementioned problem of a gradual transition in which the editing effects is superimposed to object and camera motion. In order to eliminate the motion influence on the signal behavior within the transition, motion estimation and compensation could be applied. This is computationally expensive, and, therefore, justifiable only in cases where motion information is already available, like in MPEG-compressed video [Mit97]. And, indeed, a vast number of methods have been proposed so far for designing shot-boundary detectors for MPEG-compressed video. The problem here is, however, that the detector performance may be dependent on the characteristics of the particular encoder.

2.8 REFERENCES AND FURTHER READING

[Adj97] Adjeroh D., Lee M., Orji C.: *Techniques for fast partitioning of compressed and uncompressed video,* Multimedia Tools and Applications 4(3), pp. 225-243, 1997

[Aha96] Ahanger G., Little T.D.C.: *A Survey of Technologies for Parsing and Indexing Digital Video,* Journal of Visual Communication and Image Representation, Vol. 7, No.l, pp.28-43, March 1996.

[Aig94] Aigrain P., Joly P.: *The automatic real-time analysis of film editing and transition effects and its applications,* Computer Graphics, Vol.18, No.l, pp. 93-103, January/February 1994

[Aku92] Akutsu A., Tonomura Y., Hashimoto H., Ohba Y.: *Video indexing using motion vectors,* Proceedings of SPIE Visual Communications and Image Processing, Vol. 1818, 1992

[Ala93] Alattar A.M.: *Detecting and Compressing Dissolve Regions in Video Sequences with a DVI Multimedia Image Compression Algorithm,* IEEE International Symposium on Circuits and Systems (ISCAS), Vol.1, pp 13-16, May 1993

[Ala97] Alattar A.M.: *Detecting fade regions in uncompressed video sequences,* Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol.4, pp. 3025-3028, 1997

[Ala98] Alattar A.M.: *Wipe scene change detector for segmenting uncompressed video sequences,* Proceedings of IEEE International Symposium on Circuits and Systems, Vol.4, pp. 249–252, 1998

[Arm93] Arman F., Hsu A., Chiu M.: *Feature Management for Large Video Databases,* Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases, Vol. 1908, pp. 2-12, February 1993.

[Arm94] Arman F., Hsu A., Chiu M.: *Image processing on encoded video sequences,* Multimedia Systems, Vol.1, No.5, pp. 211-219, 1994

[Bim99] Del Bimbo A.: *Visual information retrieval,* Morgan Kaufmann Publishers, Inc., 1999

[Bor93] Bordwell D., Thompson K.: *Film Art: An Introduction,* McGraw-Hill, New York 1993

[Bor96] Boreczky J.S., Rowe L.: *Comparison of video shot boundary detection techniques,* Proceedings of IS&T/SPIE Storage and Retrieval for Still Image and Video Databases IV, Vol. 2670, February 1996.

[Bor98] Boreczky J.S., Wilcox L.D.: *A Hidden Markov Model framework for video segmentation using audio and image features,* proceedings of the International Conference on Acoustics, Speech and Signal Processing, Vol. 6, pp. 3341-3744, 1998

[Bou96] Bouthemy P., Ganasia F.: *Video partitioning and camera motion characterization for content-based video indexing,* proceedings of IEEE International Conference on Image Processing, Vol.1, pp. 905-908, 1996

[Cha98] Chang S.-F., Chen W., Meng J., Sundaram H., Zhong D.: A *fully automated content-based video-search engine supporting spatiotemporal queries,* IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, pp. 602-615, 1998

[Che00] Cheong L.-F.: *Scene-based shot change detection and comparative evaluation,* Computer Vision and Image Understanding, Vol. 79, Issue 2, pp. 224-235, August 2000

[Chu99] Chung M.G., Lee J., Kim H., Song S.M.-H.: *Automatic video segmentation based on spatio-temporal features,* Korea Telecom Journal, 4(1), pp. 4-14, 1999

[Col76] Coll D.C., Choma G.K.: *Image Activity Characteristics in Broadcast Television,* IEEE Transactions on Communications, pp. 1201-1206, October 1976

[Dai95] Dailianas A., Allen R.B., England P.: *Comparisons of automatic video segmentation algorithms,* Proceedings SPIE Integration Issues in Large Commercial Media Delivery Systems, pp. 2-16, Vol. 2615, 1995

[Dem77] Dempster A., Laird N., Rubin D.: *Maximum-likelihood from incomplete data via the EM algorithm,* Journal of Roya; Statistical Society, Vol. 39, pp. 1-22, 1977

[Dud01] Duda R.O., Hart P.E., Stork D.G.: *Pattern Classification,* John Wiley & Sons, Inc., 2001

[Fer99a] Fernando W.A.C., Canagarajah C.N., Bull D.R.: *Wipe scene change detection in video sequences,* in Proceedings of IEEE International Conference on Image Processing, 1999

[Fer99b] Fernando W.A.C., Canagarajah C.N., Bull D.R.: *Fade and dissolve detection in uncompressed and compressed video sequences,* in Proceedings of IEEE International Conference on Image Processing, 1999

[For99] Ford R.M.: *Quantitative comparison of shot boundary detection metrics,* Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases, Vol. 3656, pp. 666-676, 1999

[Fur95] Furht B., Smoliar S.W., Zhang H.: *Video and Image Processing in Multimedia Systems,* Kluwer Academic Publishers 1995

[Gar00] Gargi U., Kasturi R., Strayer S.: *Performance characterization of video-shot-change detection methods,* IEEE Transactions on Circuits and Systems for Video Technology, Vol.10, No.1, February 2000

[Gu97] Gu L., Tsui K., Keightley D.: *Dissolve detection in MPEG compressed video,* IEEE International Conference on Intelligent Processing Systems (ICIPS '97), Vol.2, pp. 1692-1696, 1997

[Gui01] Guimaraes, S.J.F., Couprie, M., Leite, N.J., de Albuquerque Araujo, A.: *A method for cut detection based on visual rhythm,* Proceedings of IEEE SIBGRAPI, pp. 297-304, 2001

[Gui02] Guimaraes, S.J.F., de Albuquerque Araujo, A., Couprie, M., Leite, N.J.: *Video fade detection by discrete line identification,* Proceedings of IEEE International Conference on Pattern Recognition, pp. 1013 –1016, Vol.2, 2002.

[Haf95] Hafner J., Sawhney H.S., Equitz W., Flickner M., Niblack W.: *Efficient color histogram indexing for quadratic form distance functions,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, 1995

[Ham94] Hampapur A., Jain R., Weymouth T.: *Digital Video Segmentation,* Proceedings of ACM Multimedia'94, 1994

[Ham95] Hampapur A., Jain R., Weymouth T.: *Production model based digital video segmentation,* Multimedia Tools and Applications, Vol.1, No.1, pp.9-46, 1995

[Han02] Hanjalic A: *Shot-boundary detection: Unraveled and Resolved?,* IEEE Transactions on Circuits and Systems for Video Technology, February 2002.

[Idr97] Idris F., Panchananthan S.: *Review of Image and Video Indexing Techniques,* Journal of Visual Communication and Image Representation, 8(2), pp. 146-166, 1997

[Kas91] Kasturi R., Jain R.: *Dynamic vision,* in Computer Vision: Principles, Kasturi R., Jain R. (Eds.), IEEE Computer Society Press, Washington, 1991

[Kik92] Kikukawa T., Kawafuchi S.: *Development of an automatic summary editing system for the audio visual resources,* Transactions of the Institute of Electronics, Information and Communication Engineers, Vol J75-A, No.2, 1992

[Kim99] Kim H., Park S.-J., Lee J., Kim W.M., Song S.M.: *Processing of partial video data for detection of wipes,* in Proceedings IS&T/SPIE Conference Storage and Retrieval for Image and Video Databases VII, Vol. 3656, pp. 280-289, 1999

[Kob99] Kobla V., DeMenthon D., Doermann D.: *Special effect edit detection using VideoTrails: a comparison with existing techniques,* in Proceedings IS&T/SPIE conference Storage and Retrieval for Image and Video Databases VII, Vol. 3656, pp. 290-301, 1999

[Kuo96] Kuo T., Lin Y., Chen A., Chen S., Ni C.: *Efficient shot change detection on compressed video data,* Proceedings of the International Workshop on Multimedia Database Management Systems, 1996

[Lee94] Lee C.M., Ip M.C.: *A robust approach for camera break detection in color video sequence,* in proceedings of IAPR Workshop machine Vision Applications, Kawasaki, Japan, pp. 502-505, 1994

[Lie99] Lienhart R.: *Comparison of automatic shot boundary detection algorithms* Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases VII, Vol. 3656, January 1999

[Lie01a] Lienhart R.: *Reliable dissolve detection,* Proceedings of IS&T/SPIE Storage and Retrieval for Media Databases 2001, Vol. 4315, January 2001

[Lie01b] Lienhart R.: *Reliable transition detection in videos: A survey and practitioner's guide,* International Journal on Image and Graphics, Special Issue on Image and Video Databases, World Scientific, Singapore, Vol.1, No.3, August 2001

[Liu95a] Liu H.C., Zick G.L.: *Scene decomposition of MPEG compressed video,* Proceedings of SPIE/IS&T Symposium on Electronic Imaging Science and Technology: Digital Video Compression: Algorithms and Technologies, Vol. 2419, 1995

[Liu95b] Liu H.C., Zick G.L.: *Automatic determination of scene changes in MPEG compressed video,* Proceedings of International Symposium on Circuits and Systems (ISCAS), pp. 764-767, 1995

[Lu99a] Lu H.B., Zhang Y.J., Yao Y.R.: *Robust gradual scene change detection,* Proceedings of IEEE International Conference on image Processing, 1999

[Lu99b] Lu H.B., Zhang Y.J.: *Detecting abrupt scene changes using neural network,* proceedings of Visual Information and Information Systems, Lecture Notes in Computer Science 1614, pp. 291-298, 1999

[Lup98] Lupatini G., Saraceno C., Leonardi R.: *Scene break detection: A comparison,* research Issues in Data Engineering, Workshop on ontinuous Media Databases and Applications, pp. 34-41, 1998

[Man99] Mandal M., Idris F., Panchananthan S.: *A critical evaluation of image and video indexing techniques in the compressed domain,* Image and Vision Computing, 17, pp. 513-529, 1999

[Men95] Meng J., Juan Y., Chang S.: *Scene Change Detection in a MPEG Compressed Video Sequence,* Proceedings of IS&T/SPIE, Vol. 2419, February 1995

[Mit97] Mitchell J.L., Pennebaker W.B., Fogg C.E., LeGall D.J.: *MPEG video compression standard,* Digital Multimedia Standard Series, London, Chapman & Hall, 1997

[Nag92] Nagasaka A., Tanaka Y.: *Automatic video indexing and full-video search for object appearances,* in *Visual Database Systems II,* Eds. Knuth E. and Wegner L.M., volume A-7 of IFIP Transactions A: Computer Science and Technology, pages 113-127, North-Holland, Amsterdam 1992

[Nam00] Nam J., Tewfik A.H.: *Dissolve detection using B-splines interpolation,* IEEE International Conference on Multimedia and EXPO (ICME), Vol.3, pp. 1349-1352, 2000

[Nam01] Nam J., Tewfik A.H.: *Wipe transition detection using polynomial interpolation,* IS&T/SPIE Storage and Retrieval for Media Databases, Vol. 4315, 2001

[Ngo99] Ngo C.W., Pong T.C., Chin R.T.: *Detection of gradual transitions through temporal slice analysis,* Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 36-41, 1999

[Ots91] Otsuji K., Tonomura Y., Ohba Y.: *Video browsing using brightness data,* Proceedings of SPIE/IS&T VCIP'91, Vol.1606, 1991

[Ots93] Otsuji K., Tonomura Y.: *Projection Detecting Filter for Video Cut Detection,* Proceedings of ACM Multimedia '93, 1993

[Pap84] Papoulis A.: *Probability, Random Variables and Stochastic Processes,* McGraw-Hill, International Editions, 1984

[Pat96] Patel N.V., Sethi I.K.: *Compressed video processing for video segmentation,* Proceedings of IEEE Vision, Image and Signal Processing, 1996

[Pat97] Patel N.V., Sethi I.K.: *Video shot detection and characterization for video databases,* Pattern Recognition, Vol.30, pp. 583-592, 1997

[Red84] Redner R., Walker H.: *Mixture densities, maximum likelihood and the EM algorithm,* SIAM review, Vol. 26, pp. 195-239, 1984

[Rei68] Reisz K., Millar G.: *The technique of film editing,* Oxford, UK, Focal Press, 1968

[Ric96] Richards W., Jepson A., Feldman J.: *Priors, preferences and categorical percepts,* in Perception as Bayesian Inference, D.C. Knill, W. Richards (Eds.), pp. 93-122, Cambridge University Press, 1996

[Sal73] Salt B.: *Statistical style analysis of motion pictures,* Film Quarterly, Vol 28, pp. 13-22, 1973

[Set95] Sethi I.K., Patel N.: *A statistical approach to scene change detection,* Proceedings of SPIE/IS&T Storage and Retrieval for Image and Video Databases III, Vol. 2420, pp. 329-338, 1995

[Sey65] Seyler A.: *Probability distributions of television frame differences,* proceedings of IEEE, Vol. 53, pp. 355-366, 1965

[Sha95] Shahraray B.: *Scene change detection and content-based sampling of video sequences,* Proceedings of IS&T/SPIE Vol. 2419, pp. 2-13, February 1995

[She95] Shen K., Delp E.J.: *A fast algorithm for video parsing using MPEG compressed sequences,* IEEE International Conference on Image Processing, pp. 252-255, 1995

[She97] Shen B., Li D., Sethi I.K.: *Cut detection via compressed domain edge extraction,* IEEE Workshop on Nonlinear Signal and Image Processing, 1997

[Swa91] Swain M.J., Ballard D.H.: *Color indexing,* International Journal of Computer Vision, 7(1), pp. 11-32, 1991

[Tas98] Taskiran C., Delp E.: *Video scene change detection using the generalized trace,* IEEE International Conference on Acoustic, Speech and Signal Processing, pp. 2961-2964, 1998

[Tru00a] Truong B.T., Dorai C., Venkatesh S.: *Improved fade and dissolve detection for reliable video segmentation,* Proceedings of IEEE International conference on Image Processing (ICIP 2000), Vol. 3, 2000

[Tru00b] Truong B.T., Dorai C., Venkatesh S.: *New enhancements to cut, fade and dissolve detection processes in video segmentation,* Proceedings of ACM Multimedia 2000, November 2000

[Tse95] Tse K., Wei J., Panchanathan S.: *A scene change detection algorithm for MPEG compressed video sequences,* Proceedings of Canadian Conference on Electrical and Computer Engineering (CCECE '95), Vol. 2, pp. 827-830, 1995

[Ued91] Ueda H., Miyatake T., Yoshizawa S.: *IMPACT: An interactive natural-motion picture dedicated multimedia authoring system,* Proceedings of the CHI'91, 1991

[Vas98] Vasconcelos N., Lippman A.: *A Bayesian Video Modeling Framework for Shot Segmentation and Content Characterization,* Proceedings of CVPR '98, Santa Barbara CA, 1998

[Vas00] Vasconcelos N., Lippman A.: *Statistical models of video structure for content analysis and characterization,* IEEE Transactions on Image Processing, Vol.9, No.1, pp. 3-19, January 2000

[Wu98] Wu M., Wolf W., Liu B.: *An algorithm for wipe detection,* in Proceedings of IEEE International Conference on Image Processing, Vol. 1, pp. 893-897, 1998

[Yat99] Yates R.D., Goodman D.J.: *Probability and stochastic processes,* John Wiley and Sons, Inc., 1999

[Yeo95a] Yeo B.-L., Liu B.: *A unified approach to temporal segmentation of motion JPEG and MPEG compressed video,* Proceedings of 2nd international Conference on Multimedia Computing and Systems, pp. 81-83, 1995

[Yeo95b] Yeo B.-L., Liu B.: *Rapid scene analysis on compressed video,* IEEE Transactions on Circuits and Systems for Video Technology, Vol.5, No.6, December 1995

[Yu98] Yu H. W. Wolf: *A multi-resolution video segmentation scheme for wipe transition identification,* Proceedings of IEEE international Conference on Acoustics, Speech and Signal Processing, Vol. 5, pp. 2965-2968, 1998

[Zab95] Zabih R., Miller J., Mai K.: *A feature-based algorithm for detecting and classifying scene breaks,* Proceedings of ACM Multimedia '95, San Francisco 1995

[Zab99] Zabih R., Miller J., Mai K.: *A feature-based algorithm for detecting and classifying production effects,* Multimedia Systems, Vol. 7, pp. 119-128, 1999

[Zha93] Zhang H., Kankanhalli A., Smoliar S.W., *Automatic partitioning of full-motion video,* Multimedia Systems, Vol.1, pp. 10-28, 1993

[Zha94] Zhang H. Low C.Y., Gong Y., Smoliar S.W.: *Video parsing using compressed data,* Proceedings of SPIE Electronic Imaging, Image and Video Processing II, pp. 142-149, 1994

[Zha95] Zhang H., Low C.Y., Smoliar S.W.: *Video parsing and browsing using compressed data,* Multimedia Tools and Applications, Vol.1, No.1, pp. 91-113, 1995

Chapter 3

PARSING A VIDEO INTO SEMANTIC SEGMENTS

3.1 INTRODUCTION

Parsing a video into shots, as discussed in the previous chapter, can be considered an elementary, or *low-level,* video analysis step. The reason for such characterization is that neither this process nor the obtained results are related to the content of the video being parsed. In the process of *high-level video parsing,* however, we search for the boundaries of *semantic segments.* A semantic segment can be seen as a temporal partition of a video characterized by coherent content. While shot-boundary detection organizes video content at the syntactic level, high-level parsing provides natural semantic segmentation of video that the viewers can associate with [Wan01]. Figure 3-1 illustrates the position of semantic segments in the hierarchy of the overall video content structure.

Clearly, the notion of *content coherence* stands central in the context of high-level video parsing: semantic segments can be seen as aggregates of consecutive shots or shot parts that are linked together by content coherence. This can be illustrated by the example of a semantic segment of a movie, also referred to as *episode,* which can be defined as "a series of shots that communicate a unified action with a common locale and time" [Bog00]. Independent of possible changes in the camera angle or zoom, in the parts of the scenery, or in the persons or objects captured by the camera in different shots of an episode, we are capable of recognizing the unity of the action, the common locale and time all along the episode, just like the switch to another action, locale and time at the episode's end. The same applies to other genres as well, such as the news or documentaries. There, we are also capable of recognizing the beginning and end of a television news report or of a

thematic unit in a documentary, independent of possible differences in the content elements appearing in the individual shots of the program.

In this chapter we first explain the principle of high-level video parsing and then explore the possibilities for developing parsing methods on the basis of this principle. In particular, we concentrate on the problem of computing the values of content coherence along a video, and search for the locations of semantic segment boundaries at places where these values are found to be sufficiently low.

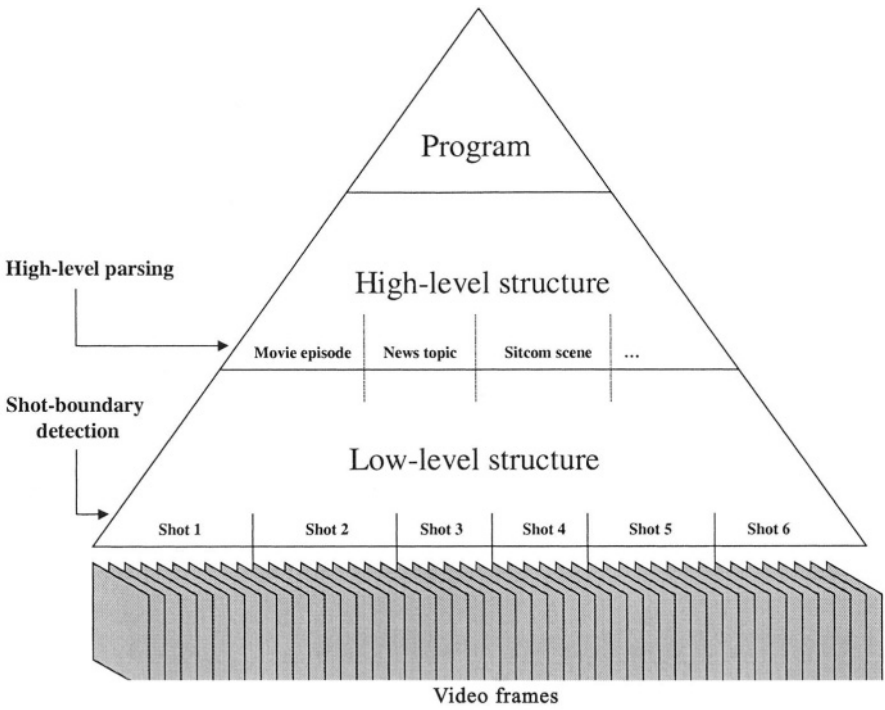

Figure 3-1. Video content structure pyramid

3.2 THE PRINCIPLE OF CONTENT COHERENCE

The idea of using the coherence principle as the basis for the development of a video parsing method was already introduced implicitly in Chapter 2. Namely, low-level video parsing can be seen as the analysis of the coherence in the visual content of a shot, that is, of the continuity of visual features along the consecutive frames of a shot. As explained in Chapter 2, a shot is taken by a single camera that, for instance, zooms in on

an object, follows a moving object or pans along a scene. Due to a limited magnitude of motion and a frame rate as high as 25 to 30 frames a second, each new frame contains a considerable portion of visual material from the previous frame. The coherence ends at the shot boundary, where the camera starts to show some other scenes or objects with – in general – drastically different visual characteristics than those in the previous shot. By computing the values of the visual content coherence along the consecutive frames of a video and by applying suitable detection mechanisms to isolate distinguishable coherence lows (i.e. sufficiently high values of visual content discontinuity $z(k)$), one can detect shot boundaries quite successfully.

Although the values of the content coherence are also based on the measurement of the relevant low-level features along a video, they are considerably more difficult to compute. Since an idea about the continuity of the content along a video can hardly be obtained by looking at one or two video frames only, the values of content coherence need to be estimated by analyzing the consistency of the features on a larger scale. This can be done by comparing, instead of single frames, larger temporal units of video, or *video clips.* These clips can be entire shots, but also the shot parts. For example, under the assumption that a movie director usually starts a new episode with a new shot, the semantic segment boundaries in movies are likely to coincide with shot boundaries. In other words, here the targeted set of semantic segment boundaries is generally a subset of shot boundaries detected along a movie. In the case of a TV news broadcast, however, the change of topics may take place in the middle of a shot (e.g. when the anchorperson moves from one topic to the next). Here, obviously, other temporal units than shots need to be considered like, for instance, the periods between the time stamps at which the anchorpersons appear, change or pause while reading. In the remainder of this chapter we will use the general term "(video) clip" when referring to the elementary temporal unit of a video that serves as the basis for the process of computing the content coherence.

Clearly, only by selecting a feature set that makes it possible to compute reliable content coherence values one can secure a good parsing performance. Basically, two questions arise regarding the feature selection process:

- Which feature set must be used to obtain a high content-coherence value when comparing the clips that belong to the same semantic segment?

- Can this feature set also reveal distinguishable low content-coherence values at semantic segment boundaries?

We formalize the process of high-level video parsing and its dependence on an appropriate set of low-level features by introducing the following definitions of *content coherence computability* and *parsable video:*

Definition 3.1

If there is a feature set ***F*** that is capable of revealing the changes in content coherence along a video, then the feature set ***F*** makes the content coherence *computable.*

Definition 3.2

If a video is generated as a concatenation of semantic segments and if the content coherence is computable in view of the feature set ***F,*** then this video is *parsable* in view of the feature set ***F.***

Definition 3.1 relates the computability of the content coherence of a video to the existence of the feature set ***F*** that we can use to compute the content coherence time curve as shown in Figure 3-2. As such, this relation can be seen as a first attempt of bridging the semantic gap introduced in Chapter 1. We will discuss the possibilities for finding an appropriate feature set ***F*** in more detail in Section 3.4.

Definition 3.2 points out that it makes sense to parse a video only if its temporal content structure allows us to do it, that is, if there are semantic segment boundaries present. As we already indicated at several places before, typical examples of such videos are news programs, movies with a clear episode-based structure and the documentaries made as series of separate thematic units. However, the process of high-level parsing can also be considered in a broader context, where the segments of a video genre which is by nature not necessarily parsable are interleaved by the segments of another video genre. This is the case, for instance, with commercial breaks in a video. There, the segments of the original video and the commercial segments can be seen as the semantic segments, and their merging points as the semantic segment boundaries.

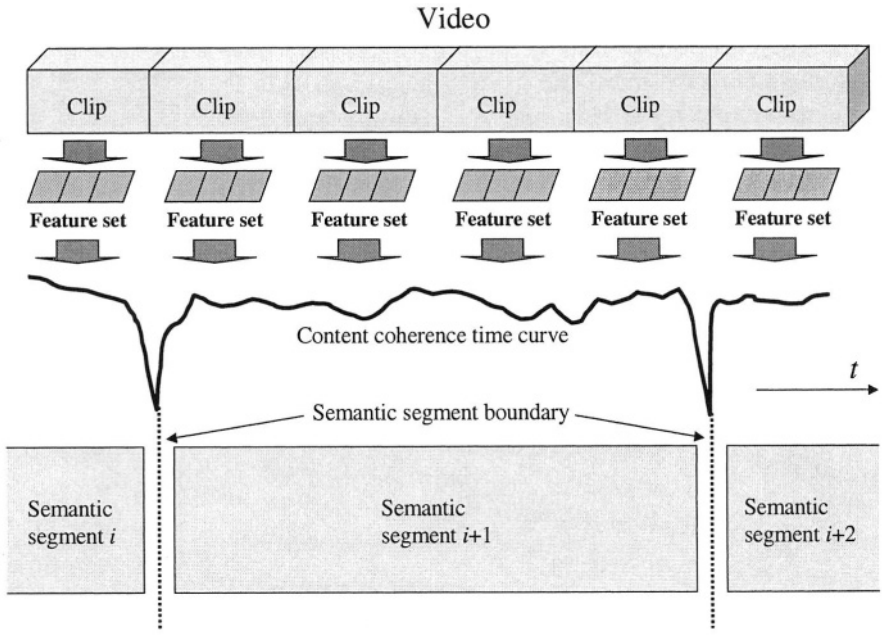

Figure 3-2. An illustration of the basic idea underlying the high-level video parsing process, Features are extracted from each clip and serve as input for the computation of the content coherence time curve. Distinguishable local minima of the content coherence time curve indicate the semantic segment boundaries.

3.3 VIDEO PARSING BASED ON THE CONTENT COHERENCE PRINCIPLE

If a video is parsable in view of the feature set ***F***, then in each semantic segment of that video we can expect to find clips that are related to each other with respect to ***F***. An important issue to note here is that these related clips are not necessarily attached to each other. Let us explain this on the example of a dialog between subjects A and B, which is typically directed by alternating the shots showing each subject, thus creating a series of shots like, for instance, ABABAB. Typical for this series of shots is that, on the one hand, all the shots A or B are very much alike in terms of their visual characteristics and often also in terms of the signal characteristics of their accompanying sound track (e.g. the voice of the subject captured by the camera). On the other hand, the difference between two consecutive shots A and B may be considerable: they show different persons, possibly against a different background, and the voices in the sound track are not the same either.

If we now let the set ***F*** consist of the features representing the visual and audio characteristics of a clip, a set of links can be established connecting the clips of this dialog as illustrated in the left part of Figure 3-3. All the clips showing the subject A can be linked together, and in the same way also all the clips showing the subject B. Ideally, the entire semantic segment will be captured by the links, and the linking process will not continue further than the last clip of the segment. Namely, under the assumption that the video is parsable in view of the feature set ***F***, the clips of the next following dialog in Figure 3-3, showing the subjects C and D, are much different from both the clips A and B from the first dialog in terms of ***F***. In the second dialog, however, again all the clips showing the subject C can be linked together, and all the clips showing the subject D as well. Ideally, the second dialog will also be captured by the links in its entirety.

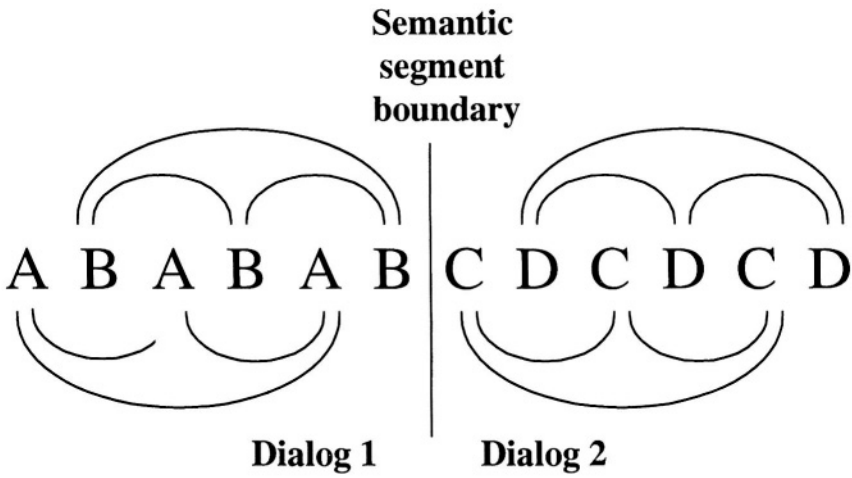

Figure 3-3. By establishing links between the clips that are related with respect to the feature set ***F*** the basis can be created for content coherence computation

Clearly, the content coherence values along a video can be computed by investigating the possibility to link the clips in the way described above. Then, high coherence values are expected at the time stamps surrounded by strong links, while low coherence can be expected where the links are weak or where no links can be established between the clips surrounding the observed time stamp.

In the following we will explore the practical possibilities for developing high-level video parsing methods on the basis of the clip-linking approach. For this purpose we introduce four illustrative ideas, namely

- time-constrained clustering,
- time-adaptive grouping,
- content recall,
- fast-forward linking.

3.3.1 Time-constrained clustering

The idea of time-constrained clustering was introduced by Yeung et al. [Yeu96a] and was used by the same authors later on to develop one of the pioneering methods for high-level video parsing via the so-called *scene transition graph* [Yeu98].

In the process of time-constrained clustering, the overlapping-links structure along a video is revealed by clustering together the clips that are similar in terms of their content and temporally not far from each other. For instance, in the example of the clip sequence shown in Figure 3-3, four clusters are expected, one for all A, B, C, or D clips. Then, all the clips in one cluster are linked together, which eventually leads to the linking scheme in Figure 3-3.

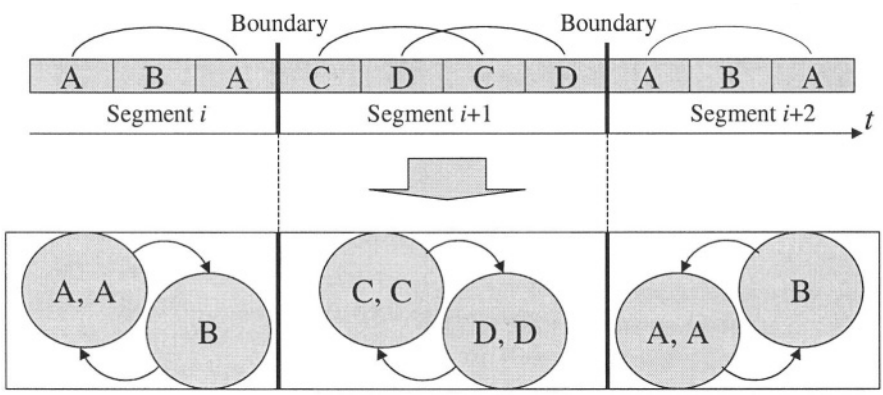

Figure 3-4. An illustration of the idea of time-constrained clustering

We emphasize again that here the clustering process is not only guided by the content similarity of the clips but by their mutual temporal locality as well. We illustrate the necessity for this time constraint by the example in Figure 3-4. The example considers a sequence of three semantic segments, with the first and the last segment containing semantically the same material. Clearly, without the time constraint, the clips A or B from the first segment would be clustered together with the corresponding clips from the third segment. As this corresponds to the propagation of the similarity links across all three segments, the two semantic segment boundaries in Figure 3-4 would be missed.

We now consider two arbitrary clips x and y in a video and denote by $d(x,y)$ and $S(x,y,\boldsymbol{V})$ their mutual temporal distance and their content similarity revealed by the feature vector $\boldsymbol{V}$, respectively. Further, let G_i be the i-th cluster of clips, T the maximum allowed temporal distance between two clips within the same cluster and $\delta>0$ the minimum required content similarity between two clips in one and the same cluster. The criteria for time-constrained clustering can now be defined by the following set of formulas:

$$S_c(x, y, \boldsymbol{F}) \geq \delta \quad \forall x, y \in G_i \tag{3.1a}$$

$$S_c(x, y, \boldsymbol{F}) < \delta \quad \forall x \in G_i, \forall y \in G_j, i \neq j \tag{3.1b}$$

Here, $S_c(x,y,\boldsymbol{F})$ is the *time-constrained similarity* between the clips x and y defined as

$$S_c(x, y, \boldsymbol{F}) = \begin{cases} S(x, y, \boldsymbol{V}), & \text{if } d(x, y) \leq T \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

with $\boldsymbol{F}$ being the overall feature set directing the clip-linking process:

$$\boldsymbol{F} = \{\boldsymbol{V}, d(.)\} \tag{3.3}$$

As illustrated in Figure 3-4, the effect of the time constraint in (3.2) becomes visible through the fact that the similar clips found further apart than allowed by the time interval T are not clustered together, but are put in separate clusters.

In view of the above, a content coherence function can be defined that relates the value of the content coherence $C(\boldsymbol{F}, t)$ at the given time stamp t to the time-constrained similarity of the clips m and n surrounding this time stamp. An example of such a function can be given as follows:

$$C(\boldsymbol{F},t) = \max_{\forall m,n \mid m<t<n} S_c(m,n,\boldsymbol{F}) \tag{3.4}$$

Clearly, the content coherence value will be higher than δ only if the time stamp t is surrounded by at least two clips belonging to the same cluster. If no elements of the same cluster can be found around the observed time stamp, then the value (3.4) will be lower than δ, as defined by (3. 1b). In this way the clustering threshold δ here serves also as the threshold which can be used to check the coherence value (3.4) upon the presence of the semantic segment boundary at the time stamp t. Due to the time constraint in (3.2), the maximization process in (3.4) only needs to take into account the clip pairs m and n that have a mutual distance smaller than T.

3.3.2 Time-adaptive grouping

The major disadvantage of time-constrained clustering is that it may suffer from discontinuities as a consequence of the "windowing effect". Namely, two clips that are sufficiently similar in terms of their content are clustered together as long as their mutual distance is shorter than T. However, when this distance becomes larger than T, the overall clip similarity is suddenly set to 0 in order to prevent that the clips merge into a cluster. Clearly, this problem makes the time-constrained clustering idea highly sensitive to the choice of the interval (window) size T.

The sensitivity of time-constrained clustering to the value of T can be reduced by using the concept of *temporal attraction* between the clips, as introduced by Rui et al. [Rui99]. The temporal attraction can be seen as a continuous and decreasing function of the temporal distance between the clips. As an example, we can use a linear function

$$Attr(x,y) = \max\left(0, 1 - \frac{d(x,y)}{L}\right) \tag{3.5}$$

which is a generalization of the function originally proposed by Rui et al. Here $d(x,y)$ is the temporal distance between the clips x and y, and L is the constant determining how fast the temporal attraction value will decrease toward 0. The behavior of the function (3.5) is illustrated in Figure 3-5.

By combining the content similarity $S(x,y,\boldsymbol{V})$ between the clips x and y, and their temporal attraction $Attr(x,y)$ we can define the *time-adaptive similarity* $S_a(x,y,\boldsymbol{F})$ of the clips x and y as

$$S_a(x, y, \boldsymbol{F}) = Attr(x, y) \cdot S(x, y, \boldsymbol{V}) \tag{3.6}$$

Here, again, $\boldsymbol{F}$ is the overall feature set directing the shot grouping process, as defined by (3.3). The function (3.6) increases (decreases) with an increasing (decreasing) content similarity of the clips and with an increasing (decreasing) temporal attraction value *Attr(x,y)*. We may also say that the function $S_a(x,y,\boldsymbol{F})$ represents the overall *attraction* between the clips *x* and *y*.

The continuous character of the similarity function (3.6) is the major factor distinguishing this function from the time-constrained similarity (3.2), which may change abruptly. Rui et al. [Rui99] introduced the clustering process based on the similarity function (3.6) as *time-adaptive grouping*. We can now compute the new content coherence value at the time stamp *t* using the expression (3.4) in which we replaced the time-constrained similarity $S_c(x,y,\boldsymbol{F})$ by the time-adaptive similarity $S_a(x,y,\boldsymbol{F})$:

$$C(\boldsymbol{F}, t) = \max_{\forall m,n \mid m<t<n} S_a(m, n, \boldsymbol{F}) \tag{3.7}$$

The expression (3.7) relates the content coherence value at the time stamp *t* to the overall attraction of the clips *m* and *n* surrounding this time stamp. Again, the threshold that evaluates the similarity $S_a(m,n,\boldsymbol{F})$ between the clips *m* and *n* and decides about their membership in the same group can directly be used as the threshold that checks the value (3.7) for the presence of the semantic segment boundary at the time stamp *t*.

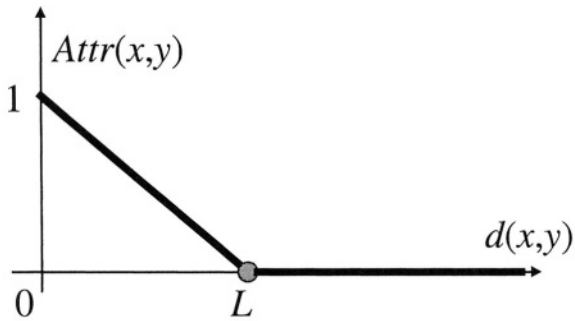

Figure 3-5. Behavior of a linear temporal attraction function (3.5)

3.3.3 Content recall

The perception of the content homogeneity in a semantic video segment may also be related to the amount of "recalled" content, that is, to the degree to which the video clips appearing later in the segment remind the viewer of the clips that appeared earlier in that segment (Kender and Yeo [Ken98]). The stronger the recall of the past video clips by the new ones, the stronger the perception of content coherence. We will refer to this further on as *content recall.*

Let us consider the clips m and n that are linked together on the basis of a strong relation between their contents with respect to the feature set $\boldsymbol{F}$. This link can be interpreted as an indication of the recall of a part of the content of m by the content of n. If we denote by $R_{mn}(\boldsymbol{F})$ the degree of recall of the clip m by the clip n, then we can obtain the value of the content recall at the time stamp t, for instance, as a sum of the values $R_{mn}(\boldsymbol{F})$ computed for all clips m older than t and all clips n newer than t [Ken98, Sun00]:

$$R(\boldsymbol{F},t) = \sum_{m<t<n} R_{mn}(\boldsymbol{F}) \tag{3.8}$$

Clearly, the value of the content recall (3.8) will vary along the sequence. As we illustrate in Figure 3-6, high values will be obtained at the time stamps surrounded by many linked clip pairs (t_0 and t_3). Somewhat lower values will result at places where less linked clip pairs are found (t_1), while very low values will occur at those time stamps that are surrounded by only few or even no links (t_2). The time stamps where extremely low values of the content recall are found are likely to separate different content units, that is, they are likely to act as semantic segment boundaries. In this sense, the behavior of the content recall along a video can be said to resemble the expected behavior of the content coherence time curve (Figure 3-2). Consequently, the content recall (3.8) can be used as a model for content coherence, that is,

$$C(\boldsymbol{F},t) = w(t) \cdot R(\boldsymbol{F},t) \tag{3.9}$$

Here, $w(t)$ is an optional normalization function the effect of which will be discussed later on.

A clear advantage of the high-level video parsing method based on the coherence function (3.9) is that the sensitivity of the parsing process to the value of a fixed threshold has disappeared. Namely, in contrast to time-constrained clustering and time-adaptive grouping, where the clip clustering/grouping threshold serves at the same time as a video parsing

threshold, here the content coherence computation is separated from the parsing process. Once the content coherence function (3.9) is computed, the values $C(\boldsymbol{F}, t)$ can be thought of as inputs in a detector that decides about the presence of a semantic segment boundary at the time stamp t and that can be developed separately and optimized in view of the required performance, complexity and available (prior) knowledge. Actually, because the coherence function $C(\boldsymbol{F}, t)$ can be understood as an (inverse) analogy to the discontinuity value time curve $z(k)$, we can approach the detection of the distinguishable coherence lows at semantic segment boundaries in a similar way as the detection of shot boundaries, for instance, by using the theory and tools of statistical optimization.

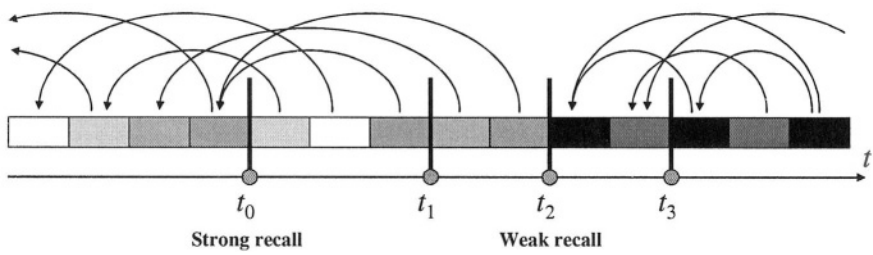

Figure 3-6. Content recall at a given time stamp of a video can be computed on the basis of the links between the clips surrounding that time stamp.

If we interpret the links between the clips in terms of content recall, then the feature set $\boldsymbol{F}$ directing the clip-linking process can be said to contain the following features [Ken98, Sun00]:

- Content similarity of the clips,
- Lengths of the clips,
- Temporal distance between the clips.

Namely, the stronger the content similarity of the clips, the stronger the recall. Further, shorter clips are less likely to be remembered than longer ones. Finally, the larger the distance between the clips, the less likely it is that they are related to each other. Consequently, the feature set $\boldsymbol{F}$ can be defined here as a vector consisting of three components:

$$\boldsymbol{F} = \{\boldsymbol{V}, l(.), d(.)\} \tag{3.10}$$

Here, again, the vector $\boldsymbol{V}$ serves as the basis for computing the content similarity $S(m,n,\boldsymbol{V})$ between the clips m and n, while $d(.)$ is their temporal distance. The component $l(.)$ represents the length of a clip.

Kender and Yeo [Ken98] compute the recall values $\boldsymbol{R_{mn}(F)}$ as a product of the content similarity $S(m,n,\boldsymbol{V})$ and the function $L(m,n)$, which takes into account the lengths and the temporal distance of the clips:

$$R_{mn}(\boldsymbol{F}) = S(m,n,\boldsymbol{V}) \cdot L(m,n) \tag{3.11}$$

The reader may notice that, basically, the expression (3.11) is an extension of expression (3.6) for the time-adaptive clip similarity. In particular, the function $L(m,n)$ can be seen as an extended version of the temporal attraction function $Attr(m,n)$.

In order to find a suitable model for the function $L(m,n)$, Kender and Yeo first investigated the decrease in the recall value between individual frames of a video as a function of their mutual distance. This decrease is modeled by the exponential function $e^{-v/B}$, where v is the distance between video frames and B is the constant determining the weakening rate of the recall with the increase in the value of v. The constant B can also be seen as the size of a *short-term memory buffer:* the longer the memory, the longer the frame from the past remains available for being recalled by a future frame. The value of $L(m,n)$ is now found by integrating the effect of the function $e^{-v/B}$ over all frame pairs in the clips m and n, that is,

$$L(m,n) = \int_{a}^{a+l(m)} \left(\int_{b}^{b+l(n)} e^{-(g-p)/B} dp \right) dg = B^2 e^{-d(m,n)/B} (1 - e^{-l(m)/B})(1 - e^{-l(n)/B}) \tag{3.12}$$

with a and b being the starting time stamps of the clips m and n, respectively. Due to the normalization, the scalar factor in (3.12) is neglected when $L(m,n)$ is replaced in (3.11) by the expression (3.12). The recall value $\boldsymbol{R_{mn}(F)}$ can now be computed as

$$R_{m,n}(\boldsymbol{F}) = S(m,n,\boldsymbol{V})\,(1 - e^{-l(m)/B})(1 - e^{-l(n)/B})\, e^{-d(m,n)/B} \tag{3.13}$$

and the content coherence value (3.9) as

$$C(\boldsymbol{F},t) = w(t) \sum_{m<t<n} \left\{ S(m,n,\boldsymbol{V})\,(1 - e^{-l(m)/B})(1 - e^{-l(n)/B})\, e^{-d(m,n)/B} \right\} \tag{3.14}$$

The normalization function *w(t)* is ideally selected such that in addition to rescaling the coherence values into the range between 0 and 1 ("no recall" and "complete recall", respectively) it also improves the quality of the obtained coherence values. Namely, an investigation of the values obtained by the formula (3.14) without normalization shows that these values are strongly biased towards maximum coherence in the middle of the video and are nearly zero at either extreme. In order to compensate for this, Kender and Yeo proposed the following normalization function:

$$w(t) = S_{\max}(\boldsymbol{V}) \cdot \sum_{m<t<n} \left\{ (1 - e^{-l(m)/B})(1 - e^{-l(n)/B})\, e^{-d(m,n)/B} \right\} \qquad (3.15)$$

Here, the value $S_{\max}(\boldsymbol{V})$ stands for the maximum possible similarity between two clips in terms of the feature set $\boldsymbol{V}$. Consequently, the function (3.15) actually corresponds to the maximum possible content recall at the time stamp *t,* given the lengths of the clips and their temporal distances.

3.3.4 Fast-forward linking

While parsing a video using the content recall method requires the computation of content coherence values (3.14) between each two consecutive video clips, the method of *fast-forward linking* (Hanjalic et al. [Han99a]) investigates the presence and strength of similarity links between the clips and computes the content coherence values in a "fast-forward" fashion for an entire series of clips. In order to explain this principle in more detail, we consider the following case set, which is also illustrated in Figure 3-7.

CASE 1: The clips k_1 and k_1+p_1 are strongly attracted to each other in terms of the feature set $\boldsymbol{F}$. Therefore, they form a linked pair and belong to the same semantic segment *E.* Consequently, all intermediate clips also belong to that semantic segment, that is,

$$[k_1, k_1 + p_1] \in E$$

with (3.16)

$$p_1 = \underset{l=1,\ldots,c}{\operatorname{argmax}} \{ Q(k_1, k_1 + l, \boldsymbol{F}) > T(k_1) \}$$

Here, c is the number of subsequent clips (the look-ahead distance) with which the current clip is compared to check their mutual attraction Q in terms of the feature set $\boldsymbol{F}$. In practice, Q may stand for the (time-adaptive) clip similarity, the recall function, or for any other criterion for establishing a semantic link between two clips. The threshold function $T(k)$ can be understood as a generalization of the thresholds introduced before in the contexts of time-constrained clustering and time-adaptive grouping and specifies the minimum value of Q required to establish a link.

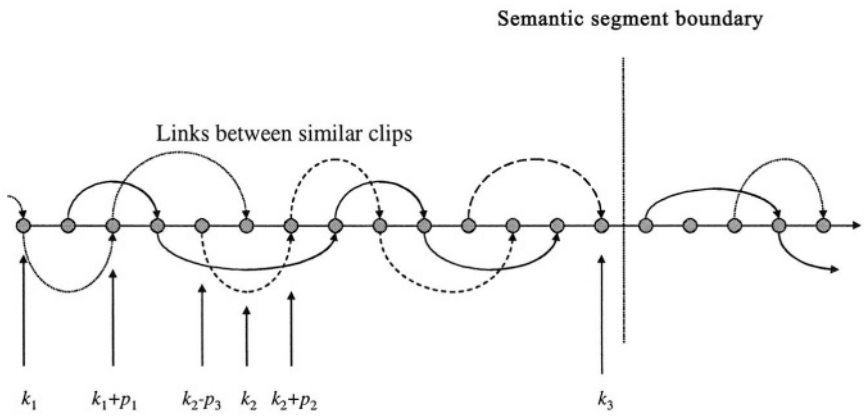

Figure 3-7. An illustration of the fast-forward linking process

CASE 2: There are no subsequent clips that are sufficiently attracted to clip k_2. However, one or more clips preceding clip k_2 may link with clip(s) following clip k_2. Then, the current clip k_2 is enclosed by a pair of clips that belong to the semantic segment E:

$$[k_2 - p_3, k_2 + p_2] \in E$$

with (3.17)

$$(p_3, p_2 > 0) = \arg\left\{ \max_{i=1,\ldots,r} \max_{l=-i+1,\ldots,c} \{Q(k_2 - i, k_2 + l, \boldsymbol{F}) > T(k_2)\} \right\}$$

Here, r is the number of clips to be considered preceding the current clip k_2 (the look-back distance).

CASE 3: If for the current clip k_3 neither (3.16) nor (3.17) is fulfilled, and if clip k_3 links with one of the previous clips, then clip k_3 is the last clip of the semantic segment *E*.

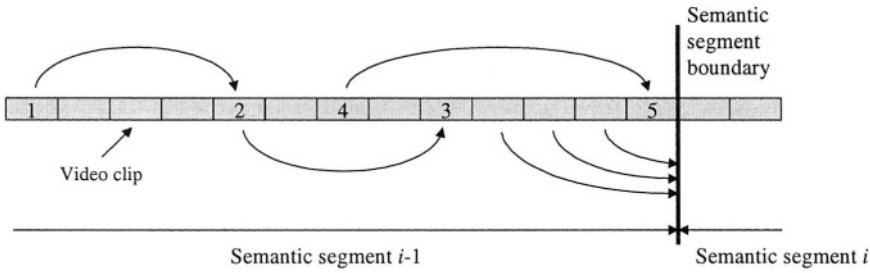

Figure 3-8. High-level parsing in a fast-forward fashion. The clips 1 and 2 can be linked together, and are by definition part of a semantic segment. Clip 3 is implicitly declared as a part of the segment since shot 4 preceding shot 3 is linked to a future clip 5. Clip 5 is at the boundary of the semantic segment since it cannot be linked to future clips, nor can any of its *r* predecessors.

To detect boundaries between semantic segments, one could, in principle, check equations (3.16) and (3.17) for all clips in the video sequence. This is computationally intensive, however, and also unnecessary. According to (3.16), if the current clip *k* is linked to clip *k+p* (e.g. the link between clips 1 and 2 in Figure 3-8), all intermediate clips automatically belong to the same semantic segment, so they need not to be checked. Only if no link can be found for clip *k* (e.g. clip 3 in Figure 3-8), it is necessary to check whether at least one of *r* clips preceding the current clip *k* can be linked with clip *k+p* (*p*>0). If such a link is found (e.g. the link between clips 4 and 5 in Figure 3-8), the procedure can continue at clip *k+p;* otherwise clip *k* marks the boundary of the semantic segment *E* (e.g. clip 5 in Figure 3-8). The procedure then continues with the next clip, that is, with the first clip of the semantic segment *E*+1.

The content coherence value $C(\boldsymbol{F},t)$ at the time stamp *t* marking the end of the considered clip *k* can now be computed as the maximum of the attraction values *Q* found, that is,

$$C(\boldsymbol{F},t)=\begin{cases} Q(k,k+p_1,\boldsymbol{F}), & \text{if (3.16) holds} \\ Q(k-p_3,k+p_2,\boldsymbol{F}), & \text{if (3.17) holds} \\ \max\limits_{i=0,..,r;\,l=-i+1,..,c} Q(k-i,k+l,\boldsymbol{F}), & \text{else} \end{cases} \tag{3.18}$$

If the equations in cases 1 and 2 are analyzed more thoroughly, it becomes clear that the possibility for fast-forward video analysis is provided by integrating the content-coherence computation and evaluation steps. In other words, through the thresholding in (3.16) and (3.17), the process of linking of video clips is made dependent on the degree of content coherence found in the series of video clips under investigation. If this degree is insufficient, no links are established and a semantic segment boundary is found. Simultaneous computation and evaluation of content coherence values was also the underlying principle of time-constrained clustering and time-adaptive grouping. Here, however, the sequential nature of the linking process makes it possible to abandon the concept of a fixed threshold and to use a more robust mechanism for detecting distinguishable local minima in the content coherence time curve. Like in the case of the parsing process based on the content recall principle, we can approach this detection problem similarly as the problem of detecting shot boundaries. Purely as an illustration of the possibility of computing the adaptive threshold function *T(k)* in practice, we mention here the threshold function that is based on a modified moving average principle [Han99a]. There, the threshold value at the clip k is computed recursively using all content coherence values obtained since the last detected semantic segment boundary, that is,

$$T(k) = a\,\overline{C}(k, N_k) \tag{3.19}$$

Here a is a fixed parameter whose value can be determined experimentally. The average $\overline{C}(k, N_k)$ is computed as

$$\overline{C}(k, N_k) = \frac{1}{N_k + 1}\left(\sum_{i=1}^{N_k} C(k-i) + C_0\right) \tag{3.20}$$

The parameter N_k denotes the number of links in the current semantic segment that have led to the current clip k, while the summation in (3.20) comprises the clips defining these links. Essentially the threshold *T(k)* adapts itself to the content inconsistencies found so far in the semantic segment. It also uses as a bias the last content coherence value C_0 of the previous semantic segment for which (3.16) or (3.17) is valid.

3.4 CONTENT SIMILARITY BETWEEN CLIPS

Independent of the approach we may choose to reveal the overlapping-links structure of a semantic video segment, the content similarity *S(x,y, V)* is one of the major criteria for establishing a semantic link between the clips x

and *y*. When computing this similarity, we need to select a feature set ***V*** that is capable of revealing the (dis)similarity of the semantic content of the clips *x* and *y*.

Already in Chapter 2 we discussed the problem of selecting an appropriate feature set for video parsing purposes in the context of shot-boundary detection. There, it was natural to assume that the feature vector used to compute the visual content discontinuity time curve *z(K)* was constant for any video. The assumption was based on the fact that this feature vector only needs to reveal the differences in visual content (and not in actual content) between video frames, for instance, the differences in their color composition. Visual content of the clips *x* and *y* can also reveal their semantic relation in some cases, like, for instance, in movies. Namely, a semantic segment of a movie, the episode, is "usually composed of a small number of interrelated shots that are unified by location or dramatic incident" [Bea94]. This means that the elements of the scenery defining the location and the foreground objects involved in the "dramatic incident" will usually appear, either completely or in part, in most of the shots of the episode. Therefore, an analysis of visual features is likely to provide a good base for establishing overlapping links between the shots of a movie episode and for revealing their semantic relation. Moreover, the boundaries between movie episodes typically coincide with a change in location and dramatic incident, which causes a change in the visual content of the shots [Ven02]. Because of the above, a visual feature vector ***V*** seems to be suitable for high-level parsing of movies.

A visual feature vector ***V*** is, however, not suitable for computing the content coherence of the clips belonging to, for instance, a TV news report. To explain this, let us consider an example news report discussing a sport event. The report typically starts with an anchorperson shot (shot showing a news reader) introducing the topic. Subsequent clips may show the city where the event took place, some aspects of the event itself, close-ups of athletes and fans, or interviews with people commenting on the event. As the clips mentioned above are deliberately chosen to show as many important aspects of the event as possible within the limited time reserved for this report, there is hardly any repetition of the visual material along the report. Clearly, clues other than visual ones need to be found that relate the clips of this semantic segment to each other and separate them from the clips of another report. In this case, the semantic links between the clips can be established, for instance, by searching for the appearance of a similar set of words in the speech accompanying the clips. The fact that all clips in a report address the same topic implies that their textual content may be rather consistent. Consequently, a text-based feature vector ***V*** may be most suitable in this case.

Although the feature vector $\boldsymbol{V}$ that optimally depicts the content similarity of video clips cannot be assumed to be unique for all video genres, in most of the parsable video genres either a visual or a text-based feature set $\boldsymbol{V}$ can be used. In sections 3.4.1 and 3.4.2 we therefore elaborate in more detail on the possibilities for computing the content similarity on the basis of a visual or text-based feature vector $\boldsymbol{V}$.

3.4.1 Visual similarity between clips

Visual similarity between clips can be computed using one of the following two major approaches:

- approach involving all frames of the clips,
- approach using visual abstracts of the clips.

A simple technique illustrating the first approach involves (a) finding the most similar pair of frames in clips x and y in terms of the feature set $\boldsymbol{V}$, (b) defining the dissimilarity $D(x,y,\boldsymbol{V})$ between the clips as the dissimilarity of this frame pair, and (c) computing the similarity $S(x,y,\boldsymbol{V})$ of the clips as the inverse of $D(x,y,\boldsymbol{V})$ [Ken98]. As in this case the problem of clip comparison is reduced to the problem of image matching, the feature set $\boldsymbol{V}$ can be selected to contain the bins of the color histogram of the frame [Yeu95a], color moments and a fractal texture dimension [Wan01], a combination of frame's texture and shape parameters [Cha95], or any other features that are typically used in the context of image matching and retrieval [Bim99]. If the dissimilarity values $D(x,y,\boldsymbol{V})$ are normalized, for instance, to the range of [0, 1], then the inversion can simply be performed as $S(x,y,\boldsymbol{V}) = 1 - D(x,y,\boldsymbol{V})$.

Instead of computing the similarity between the clips using only the best matching frame pair, we can also search for the optimal mapping between the frames of one clip and the frames of the other clip [Sha98]. Here, the value $D(x,y,\boldsymbol{V})$ can be computed as

$$D(x,y,\boldsymbol{V}) = \min_{M}\{D_M(x,y,\boldsymbol{V})\} \tag{3.21a}$$

with

$$D_M(x,y,\boldsymbol{V}) = \sum_{\forall(i,j)\in M} d(f_i^x, f_j^y) \tag{3.21b}$$

where f_i^x and f_j^y are the frames of the clip x and y, respectively, and where M is a possible one-to-one mapping between the frames of two clips.

A clear advantage of the techniques belonging to the first approach is that they require no preprocessing of the clips prior to similarity computation. However, comparing each frame of one shot with each frame of another shot may often be too complex and therefore inefficient. An alternative is offered by the techniques of the second approach. There, the visual content of a clip is first represented in a compact way, in the form of a *visual abstract,* and then this abstract is used for computing the clip similarity. Due to a large redundancy in the visual content of consecutive frames in a clip, the data set contained in a visual abstract is typically much smaller than the one used in the first approach. In the following we will discuss the techniques for computing clip similarity based on two major types of visual abstracts: *keyframes* and *mosaics.*

3.4.1.1 Clips similarity based on keyframe comparison

The problem of extracting characteristic frames – *keyframes* – from video has been addressed extensively in the past, not only for clip comparison in the context of high-level video parsing but also for the purpose of building visual interfaces for digital video [Han00, Gir01]. For the keyframe set that is to be extracted for the purpose of clip comparison we require that

- the redundancy in the visual content captured by the keyframes is minimized,
- keyframes capture all relevant aspects of the visual content of the clip.

The first requirement serves to minimize the size of the data set entering the clip comparison process. This data reduction, however, must not violate the second requirement, which secures that the extracted keyframe set is usable as a representation of the visual content of a clip. A practical consequence of the second requirement is that, in general, the number of keyframes will vary across clips depending on the magnitude of the variation in the visual content of the clips.

The techniques for reducing a video clip to a limited number of keyframes that satisfy the above requirements fall best into the category of *non-sequential keyframe extraction* (NSE) techniques, as opposed to those techniques classified as *sequential in a local* (SELC) or *global context* (SEGC) [Han00]. In order to explain the differences between these techniques, we may think of a video as a concatenation of frame series each of which is characterized by high visual-content redundancy. Depending on the dynamics of the visual content, these frame series typically include

several consecutive frames, but may also stretch to complete shots like in the example of a stationary shot showing a "talking head". Then, keyframes could be extracted from the clip in the way that the visual content redundancy is minimized in each of these frame series. In the "talking head" example, one keyframe would then be sufficient to represent the entire frame series. Consequently, keyframes are obtained as (non)equally distributed sample frames of a clip. We refer to this class of techniques as *sequential extraction in a local context* (SELC).

Since a SELC technique extracts keyframes with the objective of minimizing the visual content redundancy locally, similar keyframes may appear that are extracted from different (remote) sequence fragments. This recurrence of the similar visual material is necessary for the main application of the SELC keyframe sets, which is to provide a compact representation of the content flow along a video - a *storyboard.* However, for many other application contexts this recurrence of the visual content, and a typically high number of extracted keyframes related to it, is not required and may even be seen as a problem. This is not only the case in the context of this chapter, where a compact representation of the visual content of two clips is required for their efficient comparison, but also in keyframe-based video browsing and retrieval tools, where the abundance of keyframes makes the browsing interface too complex, slows down the interaction process and puts unnecessary large demands on storage space for keeping the keyframes.

What is required for the clip comparison and browsing applications is to minimize the number of extracted keyframes while capturing all relevant aspects of the visual content of a video. A possibility of doing this is to modify the SELC approaches by taking into account all previously extracted keyframes each time a new frame is considered for extraction. Then, a new keyframe is extracted only if it is considerably different from all other already extracted keyframes. We refer to such a technique as *sequential keyframe extraction in a global context* (SEGC). This extension has a disadvantage, however, that keyframes are compared to each other in a linear fashion: the first frame of a clip will always be extracted, as well as those frames lying in the parts of the clip with strongly varying visual content. Consequently, the extracted keyframes may capture parts of a gradual transition preceding the clip, or meaningless frames showing a fast-moving object with a considerable amount of (motion) blur. In this sense, the SEGC techniques may result in keyframes of a low representative quality.

A better alternative to the SEGC techniques are the *non-sequential keyframe extraction* (NSE) techniques, where all frames of a video are first grouped together based on the similarity of their visual content. The resulting keyframe set is then obtained by collecting the representatives of each of the groups. To do this, the NSE techniques consider the keyframe

extraction as a postprocessing step and mostly involve complex data analysis procedures. Although the SEGC techniques allow for "on-the-fly" keyframe extraction and are computationally less expensive than the NSE techniques, a relatively high complexity of NSE techniques is compensated by the fact that they are more sophisticated and capable of maximizing the representative quality of keyframes while minimizing their number.

A typical way of approaching keyframe extraction in the NSE fashion is by applying the theory and tools for *data clustering* [Jai88]. The frames of a video are first grouped into clusters, each containing the frames with similar visual content. Taking again the dialog-like video clip as an example, two clusters would be sufficient for grouping together all frames belonging to each of the content components A and B. Then, by representing each cluster by its most representative frame, a minimum-size set of two keyframes could be obtained, which optimally summarizes the given dialog video.

Since in the clustering-based approach the resulting number of keyframes is dependent on the number of clusters found in the data, the problem of finding the most suitable abstract for a given video becomes the one of finding the optimal number of clusters in which the frames of a video can be grouped. The main difficulty here is that the prior knowledge regarding the optimal number of clusters is, generally, not available and that this number needs to be determined automatically. In the approach of Hanjalic and Zhang [Han99b], the *cluster validity analysis* [Jai88] was applied to estimate the underlying cluster structure of the frames in a given video clip. First, a partitional clustering method is applied N times to all frames of a clip. The prespecified number of clusters starts at 2 and is increased by 1 each time the clustering is applied. In this way N different clustering possibilities for a video clip are obtained. The optimal number of clusters is found by computing a *cluster separation measure* $\rho(n)$ (e.g. Davies and Bouldin [Dav79]) for each clustering option, and then by searching for the distinguishable local minima in the $\rho(n)$ curve.

Now that we have an idea how to extract keyframes from the video clips that are to be compared, we could again use the feature sets $\boldsymbol{V}$ known from the context of image retrieval, and apply the same methods that were discussed before in the context of frame-to-frame clip comparison. However, as the visual content redundancy between the keyframes is minimized, the effects due to motion, zoom, occlusion or the view-point change can cause only a partial similarity, even for the keyframes of semantically related clips. This calls for more sophisticated methods for keyframe-based clip comparison that are sensitive to the appearance of a sufficient amount of similar elements of the visual content in two clips, but, at the same time, allow for possible variations in this content from one clip to another.

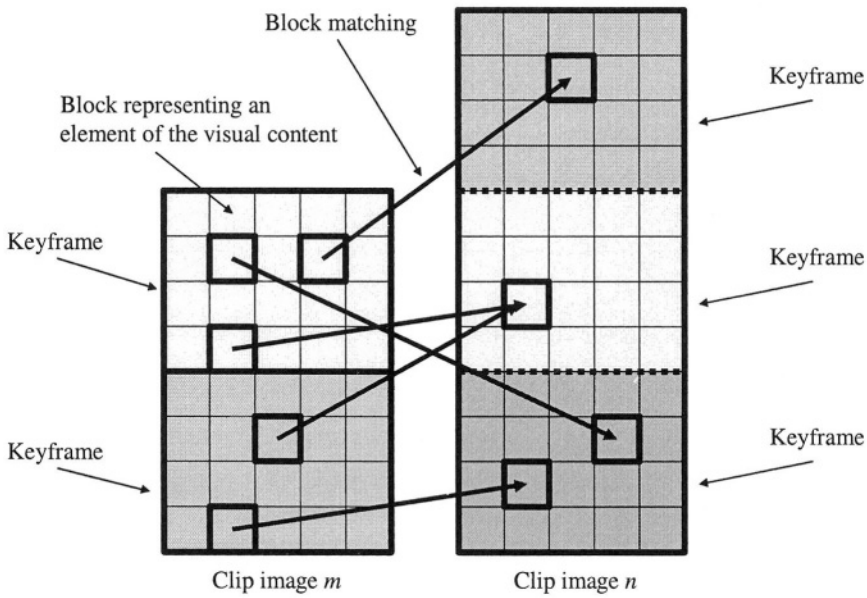

Figure 3-9. Comparing clip *m* with clip *n* by matching *HxW* pixel blocks from each keyframe of clip *m* with clip image *n.* Clip images *m* and *n* had 2 and 3 keyframes, respectively.

Hanjalic et al. [Han99a] merge the keyframes extracted per clip into one large variable-size image, *clip image,* which is then divided into the blocks of *HxW* pixels. Each block is now a simple representation of one visual-content element of the clip. Since one cannot expect an exact clip-to-clip match in most cases, and because the influence of irrelevant visual content details should be as small as possible, a feature set $\boldsymbol{V}$ is chosen that describes the *HxW* blocks globally. The vector $\boldsymbol{V}$ consists here of the components of the average color of the block. The linear *L*u*v** color space is used in order to be able to compare the average block colors using a simple Euclidean distance. Although the visual content is averaged per block, it is still possible to compare the clips in detail, as the block dimensions are sufficiently small. In order to minimize the sensitivity of the used color feature to changes in the lighting conditions in different clips, color invariance models can be applied [Sme00].

For each pair of clips *m* and *n*, with $m<n$, we would now like to find the mapping between the *HxW* blocks $\boldsymbol{b}_m$ and $\boldsymbol{b}_n$ from the clip images *m* and *n,* respectively, such that

- each block b_m in a keyframe of clip image m has a unique correspondence to a block b_n in clip image n. If a block b_n has already been assigned to a block b_m of a keyframe belonging to clip image m, no other block of that keyframe may use it. All blocks b_n are only available again when a new keyframe of clip k is to be matched. Figure 3-9 illustrates this in more detail.

- the average distance in the $L^*u^*v^*$ color space between corresponding blocks of the two shot images is minimized:

$$\min_{\forall m,n} \sum_{\forall b_m} d(b_m, b_n) \tag{3.22}$$

where $d(b_m,b_n)$ is Euclidean distance between the three-dimensional average color vectors of the blocks b_m and b_n, and where all possible block combinations are given by the first item.

As the minimization (3.22) is a problem of high combinatorial complexity, we can follow a suboptimal but more efficient approach. There, the blocks b_m of a keyframe of clip m are matched in the unconstrained way in shot image n starting with the top-left block in that keyframe, and subsequently scanning in a line-by-line fashion to its bottom-right block. If a block b_n has been assigned to a block b_m, it is no longer available for assignment until the end of the scanning path. For each block b_m the obtained match yields a minimal distance value, $d_1(b_m)$. This procedure is repeated for the same keyframe in the opposite scanning direction, i.e. from bottom-right to top-left, yielding a different mapping for the blocks b_m and a new minimal distance value for each block, denoted by $d_2(b_m)$. On the basis of these two different mappings for a keyframe of clip m and the corresponding minimal distance values $d_1(b_m)$ and $d_2(b_m)$ per block, the final correspondence and actual minimal distance $d_{\min}(b_m)$ per block is constructed using the following set of rules:

RULE 1: $d_{\min}(b_m) = d_1(b_m)$, if $d_1(b_m) = d_2(b_m)$

RULE 2: $d_{\min}(b_m) = d_1(b_m)$, if $d_1(b_m) < d_2(b_m)$ and $d_1(b_m)$ is the lowest distance value measured for the assigned block in the clip image n (one block in clip image n can be assigned to two different blocks in a keyframe of clip m: one time in each scanning direction). Otherwise, we assign $d_{\min}(b_m) = \infty$.

RULE 3: $d_{\min}(b_m) = d_2(b_m)$, if $d_2(b_m) < d_1(b_m)$ and $d_2(b_m)$ is the lowest distance value measured for the assigned block in the clip image *n*. Otherwise, we assign $d_{\min}(b_m) = \infty$.

Here, the symbol "∞" stands for a fairly large value, indicating that no meaningful best match for a block b_m could be found. The entire procedure is repeated for all keyframes of clip *m,* leading to one value $d_{\min}(b_m)$ for each block of clip image *m.*

Finally, the average of the distances $d_{\min}(b_m)$ between the *B* best-matching blocks (those with lowest $d_{\min}(b_m)$ values) in the clip image *m* is computed as the final inter-clip dissimilarity value:

$$D(m,n,V) = \frac{1}{B} \sum_{\substack{\text{sum over } B \text{ best} \\ \text{matching blocks } b_m}} d_{\min}(b_m) \tag{3.23}$$

The reason for taking only the *B* best-matching blocks is that two clips should be compared only on a global level, thus allowing for inevitable small changes in visual content within a semantic segment, which are of no influence on the content coherence. The dissimilarity value (3.23) can then be transformed into the similarity *S(m,n,V),* which was used in the expressions for computing content coherence defined before. For instance, we can first normalize the dissimilarity *D(m,n,V)* by the value of the maximum possible color difference between two blocks, and then apply the formula *S(m,n,V)* = 1 - *D(m,n,V).*

3.4.1.2 Clip similarity based on video mosaics

Extracting good keyframes to represent all important aspects of the visual content of a clip is not easy, especially in cases where this importance is difficult to determine. To illustrate this, we consider the example in Figure 3-10, showing the frames of a fictive video clip in which a foreground object ("car") passing along a complex scenery ("countryside") is tracked by the camera. As a consequence, the camera pans along the scenery, introducing new elements into the visual content with each new video frame. This makes it difficult to decide where the important aspects of the visual content appear that, when captured by keyframes, would provide relevant information for comparing this clip with another one containing the elements of the same "countryside". Another problem we encounter with the clip in Figure 3-10 is that the foreground object blocks significant parts of the background in all frames, and thus hides information that may be valuable for clip comparison.

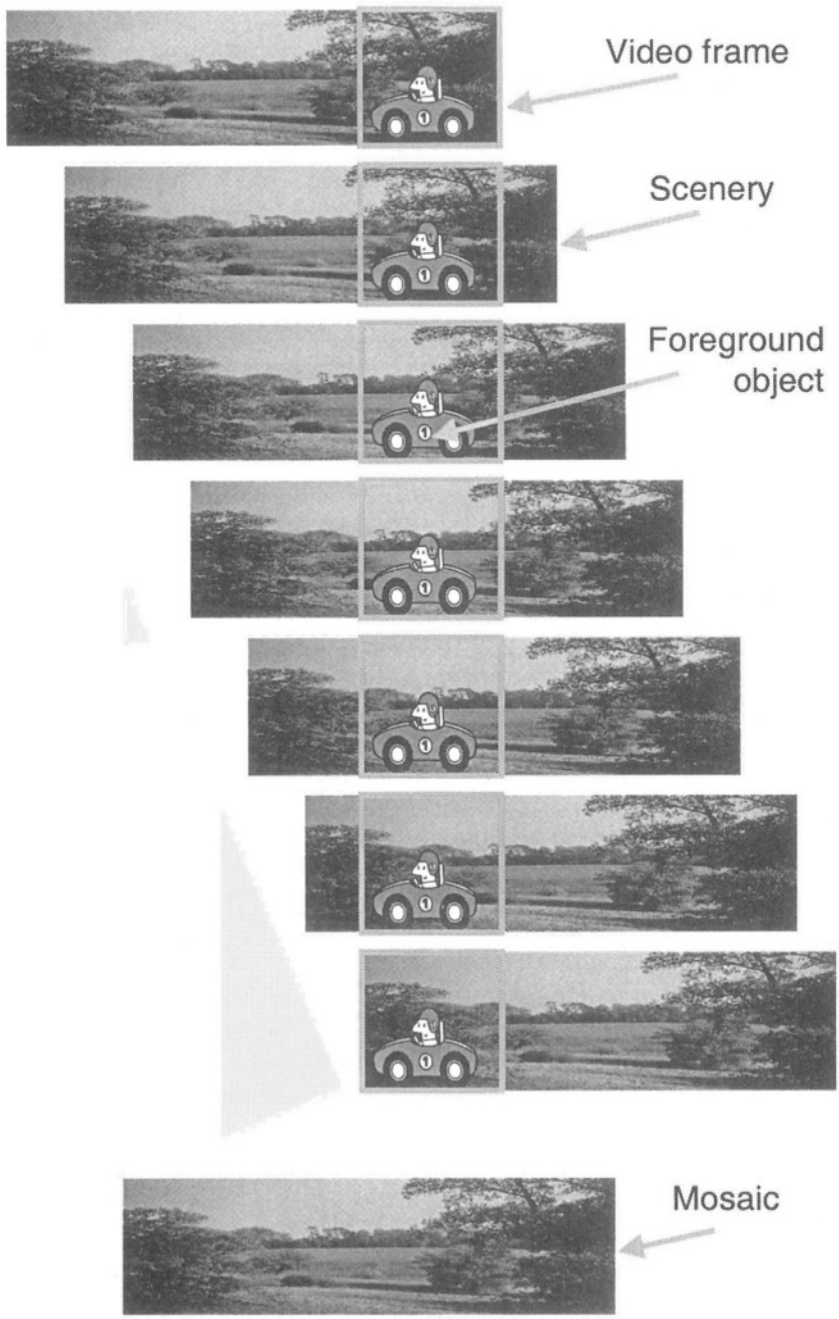

Figure 3-10. An example of a mosaic generated for a clip that is characterized by a long camera panning along a complex scenery

An alternative to the keyframe-based representation of the clip of the above example is to generate a *mosaic*. A mosaic is a single image generated from all frames of the clip. Typically, frames are aligned and projected on the mosaic image based on the information about the camera-induced displacement of the visual contents of these frames relative to the visual content of a reference frame. During this process, moving objects are usually masked out, which results in a mosaic containing only the entire static background. As shown in Figure 3-10, masking out the "car" object eliminates the effects of occlusion and reveals the complete information about the background "countryside".

In the first step of the mosaic generation process, one of the frames of the clip is selected as a reference frame. The mosaic plane will be the plane of this reference frame. Then, the (projective) transformations [Har01] are generated between successive frames of the clip, mapped to the coordinate system of the reference frame, and then used to project the frames onto the mosaic plane. The value of a pixel in the mosaic image is determined, for instance, by the median value of all pixels mapped onto this pixel [Ira96]. When generating a color mosaic, however, computing the median of each color channel could result in a color that was not originally present in the clip. To prevent this, one can first convert all frames of the clip into grey-level images while keeping the pointer from each grey-level pixel to its original color. Then, for each pixel in the mosaic, the color is selected that is pointed by the median grey-level value of all frame pixels mapped to this mosaic pixel [Ane02]. Finally, a method for outlier rejection (e.g. [Ira96]) can be applied to detect and mask out all moving objects. This is not only good for revealing the static background originally hidden behind moving objects, but it also improves the accuracy of the transformations generated between the successive frames.

Just like comparing keyframe sets, comparing the mosaics of video clips is, generally, not an easy task. Different variations in the visual contents across the clips will result in mosaics that differ in size and shape. Also, due to varying viewpoints, zooms and lighting conditions, the same scenery may appear different across several mosaics. Finally, as not every shot taken at a particular physical scene covers the whole scene, some mosaics of this scenery may contain parts that are not present in the mosaics of other shots showing the same scenery.

In view of the above, the problem of comparing the mosaics can be approached in a similar way as the problem of comparing keyframe sets, namely, by searching for a sufficient amount of similar visual material in the mosaics of different clips. We illustrate this on a simplified version of the method proposed by Aner [Ane02], where the mosaics are compared in a coarse-to-fine manner. Figure 3-11 shows the major steps of this method.

Coarse matching

In the coarse matching step, the portions in the mosaics *x* and *y* (Figure 3-11a-b) are detected that correspond to the same part of the scenery. This is done by coarsely aligning the sequences of vertical strips taken from each of the mosaics. For this purpose, a vertical strip of the mosaic *x* is compared with each vertical strip of the mosaic *y*. Since one cannot expect the visual contents of strips from different mosaics to be fully similar to each other, strips are compared block-wise. This results in a block-to-block distance matrix B where each element $B[k,l,V]$ represents the dissimilarity between the block *k* of the mosaic *x* and block *l* of the mosaic *y* (Figure 3-11c). The feature set V used to compare the blocks of the strips consists of a color histogram in the *HSI* (*H*-Hue, *S*-Saturation, *I*-Intensity) color space [Gon93]. Blocks are compared by applying the L_1 norm to their *HSI* color histograms. We then search for the sequence of matrix entries along the main diagonal in B and along the diagonals parallel to it that satisfies the following two conditions:

- It is longer than the prespecified minimum number of blocks T,
- Of all other diagonal paths longer than T, it has the lowest average of matrix entries considered.

The threshold T defines the minimum acceptable height of the matching portions in two mosaics mentioned above. Aner sets the value of T to 2/3 of the original frame height. This choice is motivated by cinematography rules [Ari76].

The diagonal sequence of entries of the matrix B satisfying the above conditions indicates the number of blocks in both strips that match well. We refer to this sequence as the "best diagonal" (Figure 3-11c). The average of the block-to-block differences $B[k,l,V]$ along the "best diagonal" is then used as the dissimilarity $Y(i,j,V)$ between vertical strip *i* in the mosaic *x* and vertical strip *j* in the mosaic *y* (Figure 3-11e). The values $Y(i,j,V)$ form the matrix where, again, the "best diagonal" is found to mark the sequences of strips in both mosaics showing the same parts of the scenery (Figure 3-11f). Here, another threshold T is used that determines the minimum acceptable width (number of consecutive vertical strips) of the matching portions in two mosaics. Aner sets this threshold equal to the width of a video frame.

The "best diagonal" in the matrix $Y(i,j,V)$ of the mosaics *x* and *y* determines the candidate matching portion in these mosaics. In order to check whether the mosaic portions marked by the "best diagonal" indeed show the same part of the scenery, a threshold found experimentally is applied to the matrix entries $Y(i,j,V)$ along the "best diagonal". If a value

*Y(i,j,**V**)* exceeds this threshold, the mosaics are considered dissimilar and are discarded from further processing.

Fine matching

Mosaic pairs that are not discarded in the previous step enter the fine matching procedure, where the cropped versions of the mosaics are used. The lengths of the best diagonals of the matrixes *B*[*k,l,**V*] and *Y(i,j,**V**)* determine the portions in both mosaics that are assumed to represent the same part of the scenery. Then, the mosaics are cropped by keeping only the matching portions and by discarding other parts of the mosaics (Figure 3-11g-h).

In the fine matching stage, the method for aligning vertical strips is applied to the cropped versions of the mosaics, with thinner strips than those used in the coarse matching step (Figure 3-11i). The result of the process is, again, the "best diagonal" containing the matrix entries that serve as the basis for computing the actual dissimilarity value *D(x,y,**V**)* for the mosaic pair considered, and for the corresponding clips *x* and *y*. Again, ways can be found for transforming the value *D(x,y,**V**)* into the corresponding similarity value *S(x,y,**V**)* that can be applied in the formulas for clip comparison we introduced before.

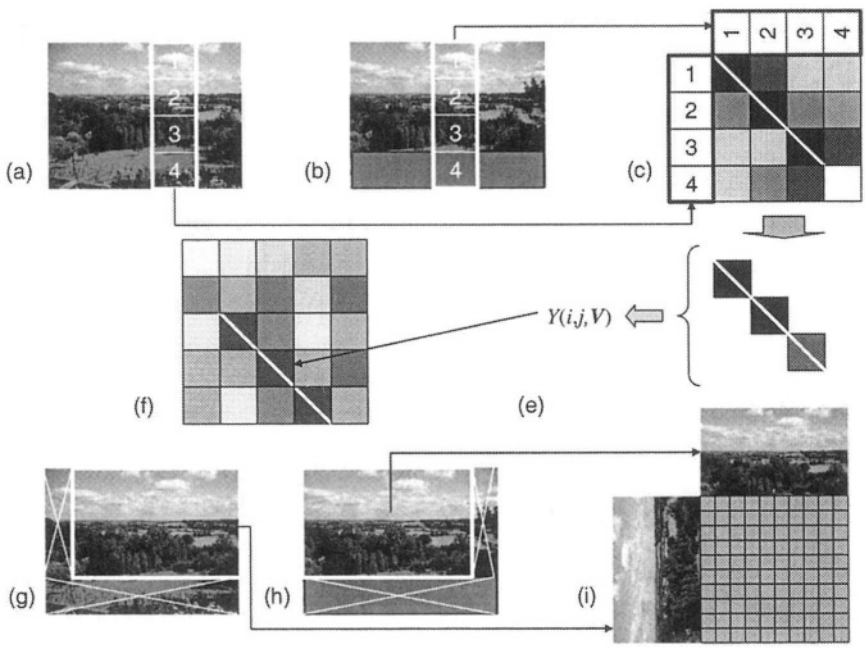

Figure 3-11. An illustration of the mosaic matching process [Ane02]

3.4.2 Similarity between clips based on accompanying text

For some video genres, like, for instance, TV news or documentaries, it is not possible to find a visual feature set $\boldsymbol{F}$ securing the computability of the content coherence. The most promising way of searching for similarity links between the clips of these programs is to analyze the topic similarity of the text accompanying these clips. This text can be obtained by transcribing the speech appearing in the audio track of the video, or by using closed captions where available.

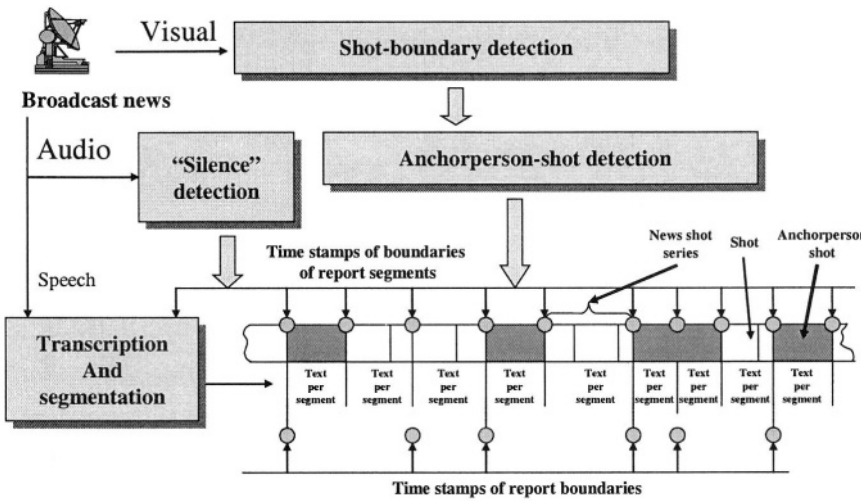

Figure 3-12. Clip detection for text-based news video parsing [Han01]

Just like in high-level video parsing based on visual content, where we exploit the visual similarity of clips to link them together, we here search for the links between clips by investigating the similarity of the topics covered by their texts. When defining the notion of the "video clip" in Section 3.2 we argued that if the semantic segment boundaries coincide with shot boundaries, shots can be used as elementary clips serving as the basic temporal units for high-level video parsing. While this is often the case in movies, a topic break in a news program can occur in the middle of an anchorperson shot, as the news reader may move from one subject to the next, for instance, after briefly pausing in reading. Moreover, a news topic usually lasts for a temporal segment consisting of several shots. Clearly, in this case temporal units other than shots need to be considered in the clip-linking process.

For determining the optimal clip boundaries for a particular video genre, domain knowledge can be used. As illustrated in Figure 3-12, we may search for the boundaries of these clips (referred to as report segments) in news programs at places where report boundaries can be expected most, that is, at the beginning and end of an anchorperson shot, or at places where the news reader makes longer pauses in speech [Han01], We will elaborate on the scheme in Figure 3-12 in more detail in Section 4.2.2.

Generally, the problem of parsing a video into semantic segments on the basis of text can be approached using the results of extensive research in the areas of text (document) analysis and text (information) retrieval in the past years and, in particular, in the field of text segmentation. Text segmentation methods aim to automatically partition texts into semantically coherent units based on, for instance, lexical cohesion [Koz93, Rey94, Hea97, Tak00, Cho00, Uti01], linguistic features [Pas97], or topicality and cue-word features [Bee99]. The techniques used for segmentation vary from *TextTiling* [Hea97], via *dotplotting* [Rey94], semantic networks [Koz93, Fer02], statistical models (HMM [Yam98], or Kullback-Leibler divergence [Bee99]), local context analysis [Pon97] to latent semantic analysis [Fol98, Cho01]. The specific approach to high-level video parsing that we have been following in this chapter – the overlapping-links approach – allows us to position our text-parsing problem even more precisely within the general area of text analysis, namely, in the context of *(story) link detection.* Here, the challenge is to determine whether two text documents can be linked by the topic they discuss [All02, Fer02]. As treating the problems of text analysis is beyond the scope of this book, we will limit the material presented in this section to the main principles only, based on which the development of the solutions to these problems can be approached.

The fundamental source of information describing the content of a text document is the set of words that is used to create that document. In order to compare the contents of two text documents, one can then simply find the overlap between the corresponding word sets and use the size of that overlap as a measure of content similarity. However, not all the words in a set are equally descriptive of the content of a document. Many of these words are also irrelevant and can therefore mislead our clip-comparison procedure. Intuitively, the reliability of a comparison of two texts is likely to improve considerably if the descriptive power of the words in each set is analyzed a priori and if the information obtained through such analysis is taken into account when computing the similarity between the sets.

The descriptive power of a word k for a given text document t can be captured in a *weight* $w(k,t)$ that is assigned to the word k and serves as argument of the function computing the text similarity. In their survey on *Simple, proven approaches to text retrieval* [Rob97] Robertson and Sparck

Jones give the basic principles of computing the weights for words in text analysis and retrieval problems. The weight for a word is, generally, a function of the following three measures:

- Word frequency,
- Collection frequency,
- Document length.

Word frequency *f(k,t)* is defined as the number of occurrences of a word *k* in a text document *t.* Repetition of words has long been recognized as a "coherence enhancer" [Tan89, Wal91, Hea97]: the more often a word occurs in a document, the more likely it is to be important for that document. The collection frequency *C(k),* also frequently referred to as the *inverse document frequency,* measures how unique a word *k* is. It is, basically, the (logarithmic) ratio of the total number of text documents considered and the number of text documents in which the word *k* occurs. If a word appears only in a few documents, its uniqueness is much higher than if that word appears in many documents. As such, the collection frequency reduces the influence - the weights - of irrelevant (common) words on the text comparison process. Finally, the length of the document needs to be taken into account in view of the intuitive argumentation that a word occurring the same number of times in a short document and in a long one is likely to be more valuable for the shorter one.

Various ideas have been proposed for combining the abovementioned basic measures together into a reliable weight *w(k,t).* We illustrate the weight computation process by formula (3.24), which has proven effective in practical trials [Rob97]:

$$w(k,t) = \frac{C(k) \cdot f(k,t) \cdot (a+1)}{a \cdot \big((1-b) + b \cdot L(t)\big) + f(k,t)} \tag{3.24}$$

Here, *L(t)* is the length of the document *t* in words, normalized by the average length of all documents considered. The tuning constants *a* and *b* determine the influence of the word frequency and document length on the total weight, respectively. Tuning is done mainly depending on the variety of words in the texts being analyzed. While for short texts the collection frequency is the most relevant component of the weight (3.24), other components become more pertinent with increasing text size and increasing variety in the corresponding word sets.

We now consider video clips *m* and *n* for which the word sets t_m and t_n are available, and denote by *K* the set of words *k* occurring in both sets with the weights $w(k,t_m)$ and $w(k,t_n)$, respectively. For comparing the clips *m* and

n we can apply the *cosine measure* [Hea97, Kau99, Cho00, Fer02], which is one of the most widely used metrics for computing the topic similarity of text documents:

$$S(m,n,V) = \cos(t_m, t_n) = \frac{\sum_{k \in K} w(k,t_m)\, w(k,t_n)}{\sqrt{\sum_{k \in K} w^2(k,t_m)} \cdot \sqrt{\sum_{k \in K} w^2(k,t_n)}} \qquad (3.25)$$

Clearly, the feature set V consists here of the weights $w(k,t)$, which reveal the semantic relations between the clips based on recurrence of relevant words. This is the most generic way of comparing texts and can be applied without restriction, regardless of the topic domains and the language of the text. However, the performance of this rather simple approach in establishing links between the clips of the same topic may not be optimal in a general case, as the topic relation between the clips is not always revealed by the occurrence of the same words in both clips. For instance, let W be the fictive set of all words that can be found for topic T by analyzing a vast number of training text documents. The set W is typically very large as one and the same topic can be covered in a variety of ways and in different contexts. Let further W_m and W_n be the subsets of W found in the texts of the video clips m and n, both covering the topic T. Clearly, if the intersection of the subsets W_m and W_n is insufficient, then the formula (3.25) will hardly be able to reveal the topic similarity of the texts in the clips m and n.

A possible cause of the problem described above is that the suffix of the word k may change in different contexts in which the word k is used. Consequently, the recurrences of the word k in different texts will not be recognized due to the variety of suffixes. A simple way to solve this problem is to apply *stemming* to all the words entering the matching process. A stemming algorithm [Lov68, Daw74, Por80, Pai90, Kro93] automatically removes the suffixes from the words, thus leaving only the word origin (stem) to be matched.

The topic-matching performance can further be improved by combining the information related to word recurrence in both texts with the information obtained by analyzing the occurrence of words that are dissimilar but mutually related, either in meaning (synonyms) or with respect to a certain topic [Hal76, Mor91]. Such information is typically contained in a *thesaurus* or in a *collocation network*. A thesaurus can be seen as a dictionary in which words with similar meanings are grouped together. In this way, a word k in the subset W_m that does not match directly any word in the subset W_n can be matched to any of the related words instead.

The term "collocation" is widely used in linguistics and means the "combination of words formed when two or more words are frequently used together in a way that sounds correct". [1] In a broader sense, we could also use the term "collocation" to describe a combination of words that frequently occur together in the text documents on one and the same topic. For example, in a text on "volcanic activity", we could expect joint occurrence of the words "volcano", "lava" and "eruption".

A collocation network can be built by using the words as nodes and with edges indicating the collocations. Each edge is marked by the *cohesion* [Ras87] value, which is used to represent the strength of the collocation of each two words. In this way, not only the word k found in the subset W_m is matched with the words in W_n but also all the words selected from the collocation network that are linked to the word k. One can, however, also be a bit more selective and select only those words from the collocation network that are linked not only to one, but to at least r words of the subset W_m [Fer98].

Each word p selected from the collocation network is assigned a weight $w(p,t)$ that will be used in the text matching process (e.g. in the cosine measure (3.25)). This weight can be computed by collecting the contributions of all words k from the text t that are linked to the word p. Ferret [Fer98] first computes the contribution of the word k to the weight of the word p as the geometric mean of the weight $w(k,t)$ of the word k and the cohesion value $coh(k,p)$ of the link between the words k and p in the collocation network. Then, the contributions of all words k are added up to compute the weight of the word p:

$$w(p,t) = \sum_i \sqrt{w(k_i,t) \cdot coh(k_i,p)} \tag{3.26}$$

Finally, we briefly address *Latent Semantic Analysis* (LSA) [Lan98] as a powerful technique for reducing the influence of word choice in evaluating the similarity of text documents. Compared to the techniques described above, the LSA has the ability to correctly infer relations between words that go much deeper (therefore the attribute "Latent Semantic") than those defined by a thesaurus or a collocation network. Namely, by simultaneously analyzing the distribution and joint occurrences of the words in all available text documents, the LSA is capable of inferring indirect semantic relations between the words: two words may be put in relation to each other even if they never occur in the same text document. In this sense, we may say that,

[1] Definition adopted from Cambridge Dictionaries Online (http://dictionary.cambridge.org), Cambridge University Press 2003

in the context of the collocation network, the LSA provides information that could be used to modify the cohesion value *coh(k,p)* leading to a more reliable weight (3.26).

Technically, the LSA is done by applying the truncated Singular Value Decomposition (SVD) to the word-document matrix $\boldsymbol{X}$. Each entry of this matrix is the original (e.g. the thesaurus- or collocation-based) weight *w(p,t)* of a word *p* in the document *t,* where *p* and *t* define the corresponding row and column of $\boldsymbol{X}$, respectively. Let the matrix $\boldsymbol{X}$ have the dimensions *i* x *j,* where *i* stands for the number of words and *j* for the number of documents in the collection. Applying the SVD to $\boldsymbol{X}$ results in the following representation:

$$\boldsymbol{X} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{Y}^T \tag{3.27}$$

where the matrices $\boldsymbol{U}$ and $\boldsymbol{Y}$ are orthogonal, where $\boldsymbol{\Sigma} = diag(\sigma_1, \ldots, \sigma_r)$ with $r = \min(i,j)$, and where σ_i are singular values. The first *r* columns of $\boldsymbol{Y}$ are called the *right singular vectors* and the first *r* columns of $\boldsymbol{U}$ the *left singular vectors.* The truncation of the SVD (3.27) is performed by approximating the matrix $\boldsymbol{X}$ using the expression

$$\boldsymbol{X} \approx \boldsymbol{U}_h\,\boldsymbol{\Sigma}_h\,\boldsymbol{Y}_h^T \tag{3.28}$$

where the matrixes $\boldsymbol{U}_h$ and $\boldsymbol{Y}_h$ consist only of the first *h* singular vectors, and where $\boldsymbol{\Sigma}_h = diag(\sigma_1, \ldots, \sigma_h)$ is the upper left *h* x *h* part of the matrix $\boldsymbol{\Sigma}$.

The truncation step changes the entries in the matrix $\boldsymbol{X}$ such that some weights *w(p,t)* increase while some other decrease compared to their original values. These changes reveal the new information for evaluating the importance of a word *p* that became available through the truncation of the SVD: by only keeping a small number of the most important singular vectors in $\boldsymbol{U}$ and $\boldsymbol{Y}$, the truncation makes the semantic relations more "visible" as the irrelevant information (noise) is greatly reduced [Li03]. Therefore, the subspace kept after the truncation is able to show the latent associative semantic relationships between words more clearly than the original document space.

3.5 AUDIO-ASSISTED VIDEO PARSING

We explained in the previous sections how visual and text features can be used to compute the similarity between video clips and to link them for the purpose of detecting semantic segment boundaries. Although it may seem straightforward to apply the same reasoning to audio features as well, these features appear not to be as reliable for high-level parsing when used alone,

compared to visual and text features. As stated by Bordwell and Thompson [Bor97], "in ordinary life sound is often simply a background for our visual attention, ... We are strongly inclined to think of sound as simply an accompaniment to the real basis of cinema, the moving images".

However, while audio may not be suitable for parsing when used alone, the quality of the parsing result obtained using visual or textual features can still be improved by using the information contained in the accompanying audio stream. For instance, the sound track of a dialog scene consisting of the interchanging speech segments originating from the persons participating in the dialog will most likely differ from the audio track of another scene where we see a nice landscape and hear only the accompanying music. Also, the change of topics in a TV news program may be characterized by the change of newsreaders or by silence intervals in speech, in which case the detection of changes in the properties of the audio track could make the detection of this topic break more robust.

An example of the possibility to use silence intervals for helping a text-based video parsing process was already given in the previous section (Figure 3-12). A typical example of where audio information may provide help in video parsing based on visual features are the ambiguous semantic segment boundaries in movies, at places where the assumption about the coherence of the visual content along an episode fails. This problem is quite realistic, for instance, due to the frequent use of "establishing" and "closing" shots to introduce and end an episode, respectively. The composition of these shots is often defined by the cinematic rules of *concentration* and *enlargement* [Wan01]. The concentration rule says that the content of an episode is to be introduced by a long distance shot, after which the camera progressively zooms in on the main objects and characters of an episode. The enlargement rule is the reverse of the concentration rule: the camera progressively zooms out from the close-ups of the main objects of the episode to show once again the general context of the episode content before switching to another episode. Clearly, due to the strong zooming actions, the visual contents of the establishing and closing shots of an episode may differ considerably from other shots of the episode. What effect this has on the video parsing approaches explained before will be illustrated on the example of the fast-forward linking approach.

We investigate a series of shots a to j, as illustrated in Figure 3-13. Let the boundary between episodes m and m+1 lie between shots e and f. We now assume that the shot e, although belonging to the episode m, has a different visual content than the rest of the shots in that episode. Consequently, the content consistency could be followed by overlapping links in m up to shot d, so that the episode boundary is found between shots d and e. If the shot e contains enough visual elements also appearing in the

episode $m+1$ so that a link can be established, e is assumed to be the first shot of episode $m+1$ instead of shot f. This results in a *displaced* episode boundary. However, if no visual similarity link can be established between shot e and any of the shots from the episode $m+1$, another episode boundary is found between shots e and f. Suppose that f is an establishing shot of the episode $m+1$, again no content-consistency link can be established from it. Another episode boundary is found between shots f and g. If the linking procedure can now be started from shot g, then g is considered to be the first shot of the new episode $m+1$. In this case, not a precise episode boundary is found but a boundary that is *spread* around the actual episode boundary, where all places where the actual episode boundary can be defined are taken into consideration. Consequently, the shots e and f are not included in the episodes. Clearly, if the actual episode boundary in this example is characterized by a strong change in the properties of the accompanying audio track, then this information could be used to remove the ambiguity resulting from the visual content analysis alone.

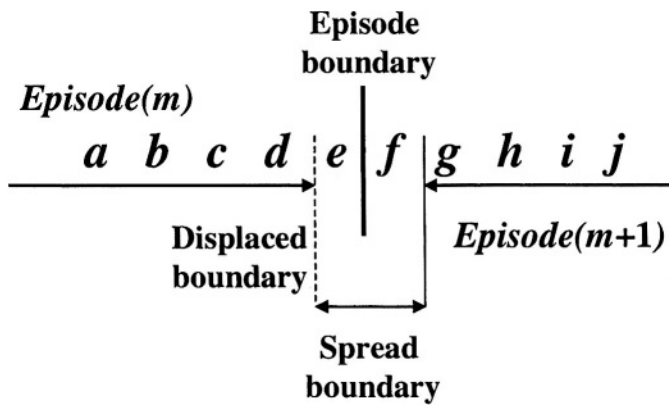

Figure 3-13. An illustration of an ambiguous episode boundary in a movie

Just like in the case of the visual and text scene boundaries introduced in Sections 3.4.1 and 3.4.2 to indicate substantial changes in the visual or textual content, respectively, we can also speak of an audio scene boundary marking a substantial change in the properties of the audio track of a video. Then, the confidence that there is a semantic segment boundary at the same place as a visual or textual scene boundary can be enlarged if an audio scene boundary is found there (or in the close vicinity) as well. In the remainder of this section we present two illustrative approaches to finding audio scene boundaries in video.

3.5.1 Audio scene boundary detection by sound classification

The detection of audio scene boundaries can be approached by classifying the elementary segments of the audio track into those characterized by, for instance, different speakers, music, environment sounds and silence [Sar97, Boc99, Jia00]. As illustrated in Figure 3-14, audio scene boundaries are then marked by the time stamps surrounded by elementary audio segments that belong to different audio classes. The modules in Figure 3-14 can be realized using a variety of existing and practically proven methods for speech/non-speech classification, speaker change detection and audio segment classification in general. As these topics are beyond the scope of this book, we only refer to relevant literature where information about the appropriate features, tools and methodologies for performing the above classifications can be found [Rab78, Pat96, Lu02b].

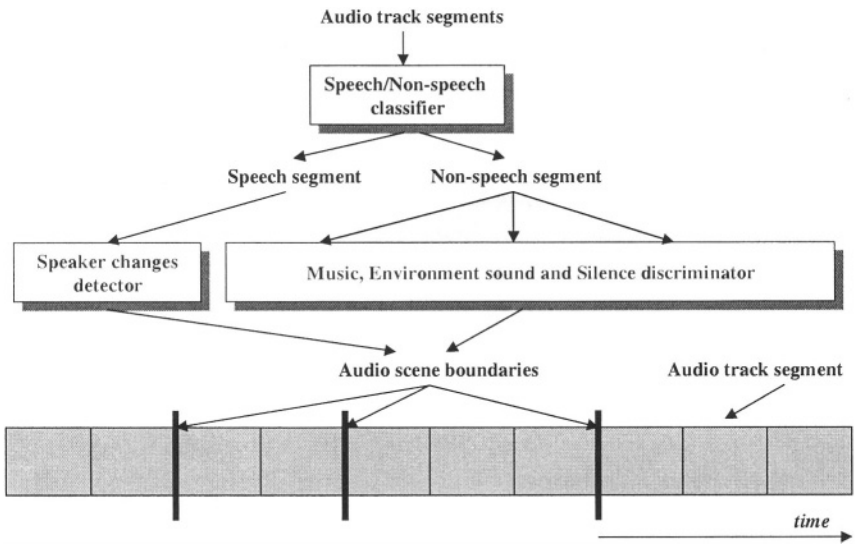

Figure 3-14. A conceptual overview of the audio scene detection method as proposed by Jiang et al. [Jia00]

Lienhart et al. [Lie99] first divide audio track of a video into segments containing foreground or background sounds. Then, a subsequent analysis step is applied where the segments containing foreground sounds are parsed using *audio cuts* that delimit elementary segment parts with coherent sound. Background-sound segments are disregarded in this second step as they are

assumed not to carry semantically relevant information. The changes of audio properties along these segments are therefore irrelevant for marking semantic segment boundaries. Finally, as illustrated in Figure 3-15, the set of audio scene boundaries is created by combining the set of time stamps corresponding to audio cuts and those corresponding to transitions between the background and foreground audio segments.

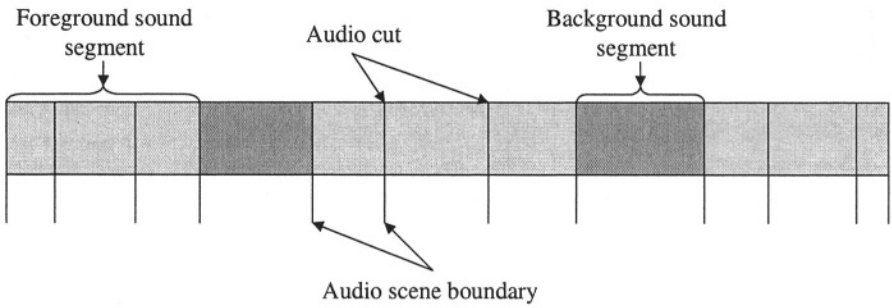

Figure 3-15. Detection of audio scene boundaries based on foreground-background sound classification [Lie99]

Technically, background-sound segments are distinguished from the foreground ones by analyzing the loudness of the audio signal. This idea is based on the assumption that whenever sounds in a video are meant to be in the background, their loudness is reduced substantially. To maximize the performance, Pfeiffer et al. [Pfe99] propose to use a loudness measure that is based on psycho-acoustic knowledge and therefore coincides with the loudness perceived by humans.

Audio cuts in a foreground sound segment are detected by performing frequency analysis in a sliding window moving along a segment. At each window position the error is computed between the actual frequency distribution and the frequency distribution predicted on the basis of signal properties at all previous positions of the sliding window since the last detected audio cut. An audio cut is detected wherever the error exceeds a predefined threshold.

3.5.2 Audio scene boundary detection by analyzing dominant sound sources

Sundaram and Chang [Sun00] introduce the concept of an audio scene as "a semantically consistent sound segment that is characterized by a few dominant sources of sound". An audio scene change occurs when the

majority of the dominant sound sources change. Various features of the audio signal can be used to characterize dominant sound sources [Pat92, Rab93, Sch97, Sri99], among which the following are proposed in [Sun00]:

- Cepstral flux,
- Multi-channel cochlear decomposition,
- Cepstral vectors,
- Low energy fraction,
- Zero-crossing rate,
- Variance of the zero-crossing rate,
- Spectral flux,
- Spectral roll-off point,
- Energy,
- Variance of energy.

We now consider the situation as illustrated in Figure 3-16. The figure shows the memory window of the length *M,* the time stamp t_0 of the audio scene boundary and the analysis window of the length *T* that has just passed the boundary and lies in its entirety in the new audio scene. The memory window can be said to contain the total information used by a listener (person) to conclude whether an audio scene change has occurred. The analysis window represents the *attention span* – the most recent data in the memory of the listener. Basically, the information contained in the attention span is compared with the information in other parts of the memory window to check for the presence of an audio scene boundary at any time stamp captured by the memory window.

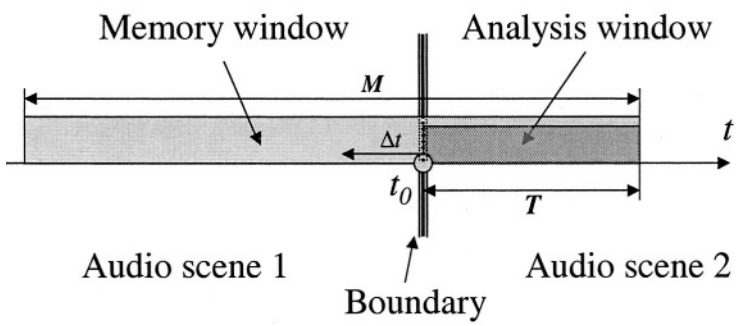

Figure 3-16. Illustration of the audio memory model [Sun00]

We first compute the feature values in each audio frame of the analysis window. These measurements result in finite time sequences of values for each feature per frame. The term "frame" stands for the chunk of data of 100ms duration. Then, for each of these time sequences, the optimal envelope fit is found, which characterizes the coarse time properties of a feature per frame. In other words, we try to find the optimal fit (in terms of an appropriate distance function) between each time sequence and one of the prespecified function types, such as constant, linear, quadratic, exponential, hyperbolic, and sum of exponentials. Each of these types, except the sum of exponentials [Sun00], can further be monotonically increasing or decreasing.

The previous two steps are repeated for all positions of the analysis window obtained by moving the window to the left by Δt and along the entire memory window. Sundaram and Chang set Δt to 1ms. For checking the presence of a boundary between two audio scenes in Figure 3-16, a function $C_i(m\Delta t)$ can be defined that determines the correlation among all time sequences measured for a feature i at the initial position of the analysis window (start at t_0) and at its position after m shifts to the left. A simple way of defining this correlation function is

$$C_i(m\Delta t) = 1 - d\left(f_i(t_0, t_0 + T), f_i(t_0 - m\Delta t, t_0 - m\Delta t + T)\right) \qquad (3.29)$$

where $f_i(t_1,t_2)$ is the function of the envelope fits of the feature i for the duration t_1 to t_2. Further, $m \in [0, M/\Delta t]$, and d is an appropriate distance function with the value range [0,1] evaluating the dissimilarity in the envelope fits at the consecutive positions of the analysis window. If there is a segment boundary at the time stamp t_0 as indicated in Figure 3-16, then the correlation function (3.29) is expected to decay rapidly as a function of m. If no boundary is present at that place, the correlation function will remain flat due to the assumed stationary dominant sound sources in an audio scene. The correlation decay for feature i can be modeled as the decaying exponential

$$Dec_i(m\Delta t) \approx e^{-b_i m} \qquad (3.30)$$

with $b_i > 0$ being the exponential decay parameter that has been selected such that the exponential function (3.30) best fits the changes in the correlation values (3.29) after m shifts of the analysis window. Since b_i determines the speed of the decay, a good indicator of the change in the audio properties at the time stamp t_0 is the sum of all coefficients b_i resulting from the approximation (3.30) for the corresponding shift $m\Delta t$ per feature i. Audio scene boundaries are then detected simply by finding distinguishable local maxima of this sum.

3.6 REMARKS AND RECOMMENDATIONS

Although the problem of high-level video parsing is considerably more complex than the one of shot-boundary detection, the existing parsing concepts that we outlined in this chapter are already capable of producing acceptable results. This can be seen in part from the comparative study of these concepts performed by Vendrig and Worring [Ven02]. Although they consider movies and situation comedies only, and base the parsing on visual features alone, their results are indicative of the usefulness of the four parsing concepts introduced in Section 3.3 from the point of view of the user, or in this specific case, a "video librarian".

In the scenario considered in [Ven02], the user has the task to manually restore the errors made by an automatic parsing method. The criterion used to evaluate an automatic parsing method is therefore made related to the effort required for manual error restoration. This evaluation criterion is defined quantitatively as

$$G = \frac{A_W - A}{A_W} \cdot 100\,\% \tag{3.31}$$

Here, the value of G can be seen as the *gain* for the user in terms of the reduction in the manual work that was originally required to parse a video but that is now done by an automatic method. The value A represents the total effort of the librarian to correct the errors of an automated parsing method, while A_W is the effort required in the worst case, that is, when the automated parsing mechanism does not detect any semantic segment boundaries in a given video. Clearly, the higher the gain G, the more useful the automated parsing mechanism.

The results reported in [Ven02] were quite good for all four concepts. For most test sequences, the method of fast-forward linking performed best, with the gain reaching 95%. A bit lower but more consistent performance over all test sequences was found for the method of time-adaptive grouping. The method of time-constrained clustering showed the best performance in terms of the total number of parsing errors. However, due to the magnitude of these errors, considerable effort was required to correct them. Therefore, the gain factors were generally lower compared to other methods, although, still, a gain of up to 77% could be reached. Last but not least, a gain of up to 91% was reached using the method of content recall, which, in view of its overall performance, is comparable to the method of time-adaptive grouping.

The good results reported above should not, however, slow down the search for ways of introducing more precision, recall and robustness in the methods for high-level video parsing. While the basic principle of content

coherence revealed by the overlapping links between elementary video clips seems to work well, one issue still deserves our attention, namely, how to successfully evaluate the content similarity of two video clips. Content similarity computation typically involves three factors:

- selecting the relevant modalities of the video (audio, visual, text) that carry the semantic information,
- selecting the feature set $\boldsymbol{F}$ securing the computability of the content coherence, and
- selecting the optimal method of representing the clip data for efficient and effective comparison.

Inspiration for further research on the above topics can be drawn from the material presented in this chapter, but also from many other ideas for high-level video parsing proposed so far and included in the literature list below.

3.7 REFERENCES AND FURTHER READING

[Ada03] Adams B.: *Where does computational media aesthetics fit?,* IEEE Multimedia, pp. 18-27, April-June 2003

[Aig95] Aigrain P., Joly P., Leplain P., Longueville V.: *Medium knowledge-based macro-segmentation into sequences,* Working notes of IJCAI Workshop on Intelligent Multimedia Information Retrieval, pp. 5-14, 1995

[All02] Allan J.: *Topic detection and tracking: Event-based information organization,* Kluwer Academic Publishers, February 2002

[Ane02] Aner A.: *Video summaries and cross-referencing,* Ph.D. thesis, Columbia University, New York, 2002

[Ari76] Arijon D.: *Grammar of the film language,* Silman-James Press, 1976

[Ari96] Ariki Y., Saito Y.: *Extraction of TV news articles based on scene cut detection using DCT Clustering,* ICIP '96, Vol. 3, pp. 847-850, Lausanne CH, 1996

[Bea94] Beaver F.: *Dictionary of film terms,* Twayne Publishing, New York, 1994

[Bee99] Beeferman D., Berger A., Lafferty J.: *Statistical models for text segmentation,* Machine Learning, 34/1, pp. 177-210, 1999

[Bim99] Del Bimbo A.: *Visual information retrieval,* Morgan Kaufmann Publishers, Inc., 1999

[Boc99] Boccignone G., De Santo M., Percannella G.: *Joint audio-video processing of MPEG encoded sequences,* proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS), 1999

[Boc00] Boccignone G., De Santo M., Percannella G.: *A system for parsing MPEG videos,* IS&T/SPIE Internet Imaging, 2000

[Bog00] Boggs J.M., Petrie D.W.: *The art of watching films,* 5th ed., Mountain View, CA: Mayfield 2000

[Bor97] Bordwell D., Thompson K.: *Film Art: An Introduction,* McGraw-Hill, New York, 1997

[Cha95] Chang S.-F., Smith J.R.: *Extracting multidimensional signal features for content-based visual query,* SPIE Symposium on Visual Communications and Signal Processing, pp. 995-1006, 1995

[Chi00] Chiu P., Girgensohn A., Polak W., Rieffel E., Wilcox L.: *A genetic algorithm for video segmentation and summarization,* Proceedings of IEEE International Conference on Multiemdia and EXPO (ICME), Vol. 3, pp. 1329-1332, 2000

[Cho00] Choi F.: *Advances in domain independent linear text segmentation,* NAACL '00, pp.26-33, 2000

[Cho01] Choi F., Wiemer-Hastings P., Moore J: *Latent semantic analysis for text segmentation,* proceedings of the NAACL '01, pp. 109-117, 2001

[Cor98] Corridoni J.M., Del Bimbo A.: *Stuctured representation and automatic indexing of movie information content,* Pattern Recognition, 31(12), pp. 2027-2045, 1998

[Dav79] Davies D.L., Bouldin D.W.: *A cluster separation measure,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, pp. 224-227, April 1979

[Daw74] Dawson J.L.: *Suffix removal and word connation,* ALLC Bulletin, 2(3), pp. 33-46, 1974

[Fer98] Ferret O.: *How to thematically segment texts by using lexical cohesion?,* ACL-COLING '98, pp. 1481-1483,1998

[Fer02] Ferret O.: *Using collocations for topic segmentation and link detection,* COLING '02, 2002

[Fis99] Fiscus J., Doddington G., Garofolo J., Martin A.: *MSTS's 1998 topic detection and tracking evaluation,* DARPA Broadcast News Workshop, 1999

[Fol98] Foltz P.W., Kintsch W., Landauer T.K.: *The measurement of textual coherence with Latent Semantic Analysis,* Discourse Processes, 25, 2&3, pp. 285-307, 1998

[Fur95] Furht B., Smoliar S.W., Zhang H.: *Video and Image Processing in Multimedia Systems,* Kluwer Academic Publishers, 1995

[Gir0l] Girgensohn A., Boreczky J., Wilcox L.: *Keyframe-based user interfaces for digital video,* IEEE Computer, September 2001

[Gon93] Gonzales R.C., Woods R.E.: *Digital image processing,* Addison Wesley, 1993

[Gun97] Gunsel B., Fu Y., Tekalp A.M.: *Hierarchical temporal video segmentation and content characterization,* in Multimedia Storage and Archiving Systems II, proceedings of SPIE, Vol. 3229, pp. 46-56, 1997

[Hal76] Halliday M.A.K., Hasan R.: *Cohesion in English,* Lohman, London 1976

[Han98] Hanjalic A., Lagendijk R.L., Biemond J.: *Template-based detection of anchorperson shots in news programs,* Proceedings of IEEE International Conference on Image Processing (ICIP), 1998

[Han99a] Hanjalic A., Lagendijk R.L., Biemond J.: *Automated high-level movie segmentation for advanced video-retrieval systems,* IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No.4, pp. 580-588, June 1999

[Han99b] Hanjalic A., Zhang H.-J.: *An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis,* IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No.8, December 1999

[Han00] Hanjalic A., Langelaar G.C., van Roosmalen P.M.B., Biemond J., Lagendijk R.L.: *Image and video databases: restoration, watermarking and retrieval,* Elsevier Science, Amsterdam 2000

[Han01] Hanjalic A., Kakes G., Lagendijk R.L., Biemond J.: *Indexing and retrieval of TV broadcast news using DANCERS,* Journal of Electronic Imaging, 10(4), pp. 871-882, October 2001

[Har01] Hartley R., Zisserman A.: *Multiple view geometry in computer vision,* Cambridge University Press, 2001

[Hea97] Hearst M.: *TextTiling: Segmenting text into multi-paragraph subtopic passages,* Computational Linguistics, 23/1, pp. 33-64, 1997

[Ira96] Irani M., Anandan P., Bergenand J., Kumar R., Hsu S.: *Efficient representation of video sequences and their applications,* Signal Processing: Image Communication, Volume 8, 1996

[Jai88] Jain A.K., Dubes R.C.: *Algorithms for clustering data,* Engelwood Cliffs, NJ, Prentice Hall, 1988

[Jia00] Jiang H., Lin T., Zhang H.-J.: *Video Segmentation with the assistance of audio content analysis,* IEEE International Conference on Multimedia and Expo (ICME2000), 2000

[Kau99] Kaufmann S.: *Cohesion and collocation: Using context vectors in text segmentation,* ACL '99, pp. 591-595, 1999

[Ken98] Kender J.R., Yeo B.-L.: *Video scene segmentation via continuous video coherence,* Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 1998

[Koz93] Kozima H.: *Text segmentation based on similarity between words,* Proceedings of the 31st Annual Meeting of the Association for Compuational Linguistics, 1993

[Kro93] Krovetz R.: *Viewing morphology as an inference process,* Proceedings of the 16th ACM SIGIR Conference, pp. 191-202, 1993

[Kwo00] Kwon Y.-M., Song C.-J., Kim I.-J.: *A new approach for high level video structuring,* Proceedings of the IEEE International Conference on Multimedia and EXPO (ICME), Vol. 2, pp/ 773-776, 2000

[Lan98] Landauer, T. K., Foltz, P. W., Laham, D.: *Introduction to Latent Semantic Analysis. Discourse Processes,* Vol. 25, pp. 259-284,1998

[Lee00] Lee S.-Y., Lee S.-T,, Chen D.-Y.: *Automatic video summary and description,* Lecture Notes in Computer Science, Vol. 1929, pp. 37-48, Springer Verlag, Berlin 2000

[Li03] Li D., Dimitrova N., Li M., Sethi I.K.: *Multimedia content processing through cross-modal association,* Proceedings of ACM Multimedia '03, Berkeley 2003

[Lie99] Lienhart R., Pfeiffer S., Effelsberg W.: *Scene determination based on video and audio features,* Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS), 1999

[Lin00] Lin T., Zhang H.J.: *Automatic video scene extraction by shot grouping,* Proceedings of International Conference on Pattern Recognition (ICPR), 2000

[Lov68] Lovins J.B.: *Development of a stemming algorithm,* Mechanical Translation and Computational Linguistics, 11, pp. 22-31, 1968

[Lu02a] Lu X., Y.-F. Ma, H. Zhang, L. Wu, *An integrated correlation measure for semantic video segmentation,* Proceedings of IEEE International Conference on Multimedia and Expo, ICME, Lausanne, Switzerland, August, 2002.

[Lu02b] Lu L., Zhang H.-J., Jiang H.: *Content analysis for audio classification and segmentation,* IEEE Transactions on Speech and Audio Processing, Vol. 10, No.7, October 2002

[Moh96] Mohan R.: *Text-Based Search of TV News Stories,* Proceedings of SPIE Conference on Multimedia Storage and Archiving Systems SPIE, Boston, MA, November 1996 pp. 2-13.

[Mor91] Morris J., Hirst G.: *Lexical cohesion computed by thesaural relations as an indicator of the structure of text,* Computational Linguistics, 17(1), pp. 21-48, 1991

[OCo01] O' Connor N., Czirjek C. and al.: *News Story Segmentation in The Físchlár Video Indexing System,* Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 7-10, 2001

[Pai90] Paice C.P.: *Another stemmer,* Department of Computing, Lancaster University, UK, 1990

[Pas97] Passoneau R.J., Litman D.J.: *Discourse segmentation by human and automated means,* Computational Linguistics, 23(1): 103-139, 1997

[Pat96] Patel N.V., Sethi I.K.: *Audio characterization for video indexing,* IS&T/SPIE Electronic Imaging: Storage and Retrieval for Image and Video Databases IV, Vol. 2670, pp. 373-384, 1996

[Pat92] Patterson R.D., Robinson K., Holdsworth J., McKeown D., Zhang C., Allerhand M.H.: *Complex sounds and auditory images,* in Auditory Psychology and Perception, (Eds.) Y. Cazals, L. Demany, K. Horner, Pergamon, Oxford, 1992.

[Pfe99] Pfeiffer S.: *The importance ofperceptive adaptation of soundfeatures for audio content processing,* IS&T/SPIE Electronic Imaging: Storage and Retrieval for Image and Video Databases VII, 1999

[Pon97] Ponte J., Croft W.B.: *Text segmentation by topic,* In proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, pp. 120-129, 1997

[Por80] Porter M.F.: *An algorithm for suffix stripping,* Program, 14, No.3, pp. 130-137, July 1980

[Rab93] Rabiner L.R., Huang B.H.: *Fundamentals of speech recognition.* Prentice Hall, 1993

[Rab78] Rabiner L.R., Shafer R.W.: *Digital processing of speech signals,* Prentice Hall, 1978

[Ras87] Raskin V., Weiser I.: *Language and writing: applications of linguistics to thetoric and composition,* ABLEX Publishing Corporation, Norwood, NJ, 1987

[Rey94] Reynar J.C.: *An automatic method of finding topic boundaries,* Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, 1994

[Rob97] Robertson S.E., Sparck Jones K.: *Simple proven approaches to text retrieval,* Technical Report TR356, Cambridge University, Computer laboratory, 1997

[Rui98] Rui Y., Huang T.S., Mehrotra S.: *Exploring video structure beyond the shots,* Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS), pp. 237-240, 1998

[Rui99] Rui Y., Huang T.S., Mehrotra S.: *Constructing table-of-content for videos,* Multimedia Systems, Special Section on Video Libraries, 7(5), pp. 359-368, 1999

[Sah99] Sahouria E., Zakhor A.: *Content analysis of video using principal components,* IEEE Transactions on Circuits and Systems for Video Technology, 9(8), pp. 1290-1298, 1999

[Sar97] Saraceno C., Leonardi R.: *Audio as a support to scene change detection and characterization of video sequences,* proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1997

[Sch97] Scheirer E., Slaney M.: *Construction and evaluation of a robust multifeature speech/music discriminator,* Proceedings of (ICASSP), 1997

[Sha98] Shan M.-K., Lee S.-Y.: *Content-based video retrieval based on similarity of frame sequence,* Proceedings of the International Workshop on Multi-Media Database Management Systems, pp 90 – 97, August 1998

[Sko72] Skorochod'ko E.: *Adaptive method of automatic abstracting and indexing,* In C. Freiman (Eds.): Information Processing 71: Proceedings of the IFIP Congress 71, pp. 1179-1182, North-Holland Publishing Company, 1972

[Sme00] Smeulders A.W.M., Worring M., Santini S., Gupta A., Jain R., *Content-based image retrieval at the end of the early years.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12): 1349--1380, 2000

[Sri99] Srinivasan S., Petkovic D., Ponceleon D.: *Towards robust features for classifying audio in the CueVideo system,* Proceedings of the seventh ACM international conference on Multimedia, October 1999

[Sun00] Sundaram H., Chang S.-F.: *Determining computable scenes in films and their structures using audio-visual memory models,* Proceedings of the 8th ACM Multimedia Conference, 2000

[Tak00] Takao S., Ogata J., Ariki Y.: *Topic segmentation of news speech using word similarity,* Proceedings of the ACM Multimedia Conference, 2000

[Tan89] Tannen D.: *Talking voices: repetition, dialogue and imagery in conversational discourse,* Studies in International Sociolinguistics 6, Cambridge University Press, 1989

[Tru02] Truong B.T., Venkatesh S., Dorai C.: *Neighborhood coherence and edge-based approach for scene extraction in films,* proceedings of IEEE International Conference on Pattern Recognition (ICPR), 2002

[Uti01] Utiyama M., Isihara H.: *A statistical model for domain-independent text segmentation,* ACL '01, pp. 491-498, 2001

[Ven01] Vendrig J., Worring M. Smeulders A.W.M.: *Model based interactive story unit segmentation,* IEEE International Conference on Multimedia and Expo (ICME), pages 1084-1087, August 22-25, 2001

[Ven02] Vendrig, J., Worring, M.: *Systematic evaluation of logical story unit segmentation* IEEE Transactions on Multimedia, Vol. 4, No. 4, pp. 492 –499, Dec. 2002

[Ven03] Vendrig, J.; Worring, M.: *Interactive adaptive movie annotation,* IEEE Multimedia, Vol. 10, No. 3, pp. 30 –37, July-Sept. 2003

[Ven00] Veneau E., Ronfard R., Bouthemy P.: *From video shot clustering to sequence segmentation,* Proceedings of International Conference on Pattern Recognition (ICPR), Vol. 4, pp. 254-257, 2000

[Wal91] Walker M.: *Redundancy in collaborative dialogue,* In J. Hirschberg, D. Litman, K. McCoy, C. Sidner (Eds.): AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation, Pacific Grove, CA, 1991

[Wan01] Wang J., Chua T.-S., Chen L.: *Cinematic-based model for scene boundary detection.* Proceedings of MMM' 2001 (Multimedia Modeling Conference), Amsterdam, Netherlands, Nov 2001. pp 3-18

[Yam98] Yamron J., Carp I., Gillick L., Lowe S., van Mulbregt P.: *A hidden Markov model approach to text segmentation and event tracking,* Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1998

[Yeu95a] Yeung M., Liu B.: *Efficient matching and clustering of video shots,* Proceedings of the International Conference on Image Processing (ICIP), pp 23-26, 1995

[Yeu95b] Yeung M., Yeo B.-L., Wolf W., Liu B.: *Video browsing using clustering and scene transitions on compressed sequences,* Proceedings of Multimedia Computing and Networking 1995, Vo. SPIE 2417, pp. 399-413, 1995

[Yeu96a] Yeung M., Yeo B.-L.: *Time-constrained clustering for segmentation of video into story units,* Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 375-380, 1996

[Yeu96b] Yeung M., Yeo B.-L., Liu B.: *Extracting story units from long programs for video browsing and navigation,* Proceedings of the IEEE international Conference on Multimedia Computing and Systems, pp. 296-305, 1996

[Yeu97] Yeung M., Yeo B.-L.: *Video visualization for compact presentation and fast browsing of pictorial content,* IEEE Transactions on Circuits and Systems for Video Technology, Vol.7, No.5, pp. 771-785, 1997

[Yeu98] Yeung M., Yeo B.-L., Liu B.: *Segmentation of video by clustering and graph analysis,* Computer Vision and Image Understanding, 71(1), pp.94-109, 1998

[Zha95] Zhang H.J., Tan S.Y., Smoliar S.W., Yihong G.: *Automatic parsing and indexing of news video,* Multimedia Systems, 2(6), pp. 256-266, 1995

Chapter 4

VIDEO INDEXING AND ABSTRACTION FOR RETRIEVAL

4.1 INTRODUCTION

The parsing techniques discussed in the previous chapters have the task to reveal the temporal structure (distribution) of the content in a parsable video and mark the boundaries of the temporal content units that may be interesting for retrieval later on. Such units can also be found in a non-parsable video like, for instance, the segments showing suspicious human behavior in a continuous surveillance video recording.

In order to make the content of an arbitrary temporal video segment (video clip) easily accessible to the user, two additional video content analysis steps are required, namely

- Video indexing,
- Video abstraction.

The *video indexing* step involves searching in a given video clip for the appearance of the content that is described by a *content label.* Typically, a label of this kind is prespecified by the user and represents the type of content the user is interested in having extracted from video. For instance, the user may want to access all reports in a news program where the topics labeled as "Parliament", "United Nations", "Amsterdam" or "Foreign politics" are discussed. In a movie, the user may want to search for all episodes labeled as "action" or "romance", and in a wildlife documentary for all scenes with the content described by the labels "hunt" or "running lion". If the content specified by the label is found in a video clip, then this clip is *indexed (annotated, labeled)* by this label and becomes easily accessible to

the user. For instance, all episodes containing a car chase in a collection of action movies can quickly be retrieved by simply "clicking" on the label "Car chase" in an adequate user interface.

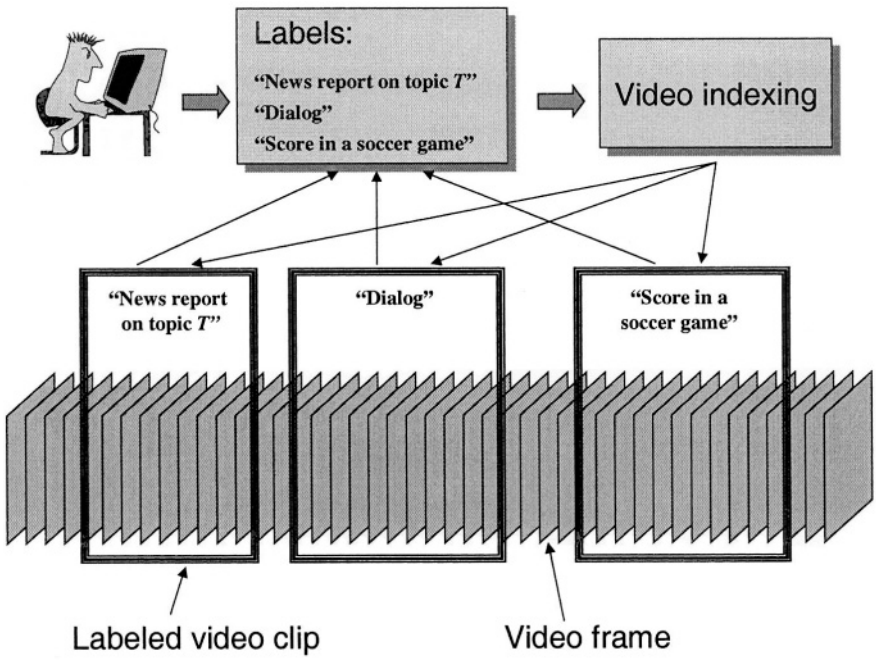

Figure 4-1. Video indexing and retrieval using predefined content labels

The techniques for automatically assigning a content label to a video clip are referred to as *video indexing* or *annotation techniques.* Figure 4-1 illustrates the process of video indexing on the basis of three labels "News report on topic *T*", "Dialog" and "Score in a soccer game". After being specified by the user, these labels are first used to index the corresponding clips in the video collection, and then used again later on in the process of retrieving these clips from the collection.

An interface employing index terms (content labels) predefined by the user is most intuitive for retrieving the desired pieces of a video collection. A simple example of such an interface is shown in Figure 4-2 and enables topic-based interaction with a broadcast news archive [Han01]. On the left side, the labels are listed that represent the topic categories being of interest to the user. Each label provides links to all news reports found in the archive

that are classified under the corresponding topic category using an appropriate indexing method. On the right side of the interface the space is left for displaying the retrieval results.

Clearly, in the case of a large news archive, an index term in Figure 4-2 may point to a vast number of news reports not all of which are of interest to the user at the given moment. Efficiently finding the report(s) of interest in this case can be made possible, for instance, by predefining the content labels more specifically prior to the indexing process, or by generating a hierarchical tree of index terms that is capable of guiding the user from more general semantic concepts (e.g. "Olympic Games"), through more specific ones (e.g. "Winter Olympic Games 1984"), to very detailed ones (e.g. "Alpine skiing, Ladies, Slalom, First Race, Skier A"). Generating such hierarchical trees is, however, not an easy task in view of an enormous number of different possible semantic concepts at each tree level, but also in view of a high number of complex indexing steps required for the defined labels. Moreover, browsing through such extensive trees of terms is not necessarily the most user-friendly way of interacting with a video collection, especially for the users who were not involved in the process of defining the terms and for whom therefore not all the terms may be meaningful.

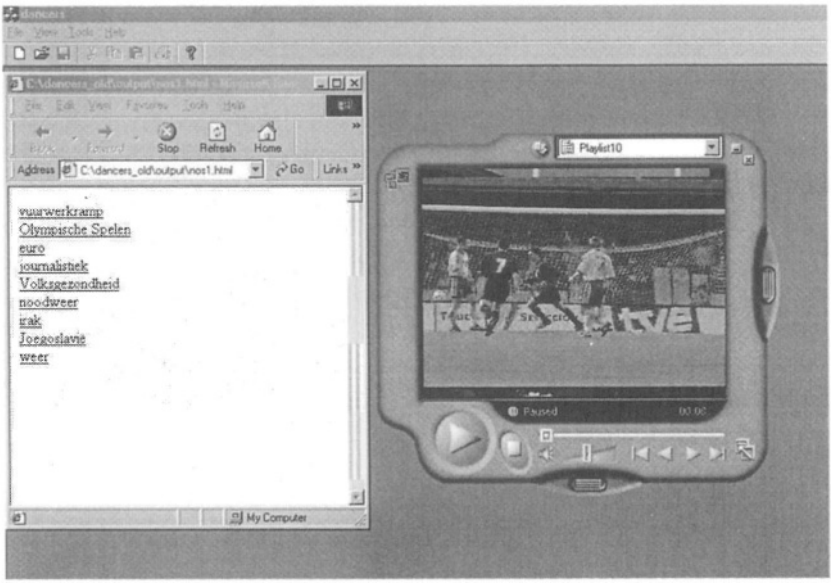

Figure 4-2. An example of a simple user interface for interacting with a broadcast news archive on the basis of content labels predefined by the user and originally employed for indexing purposes [Han01]

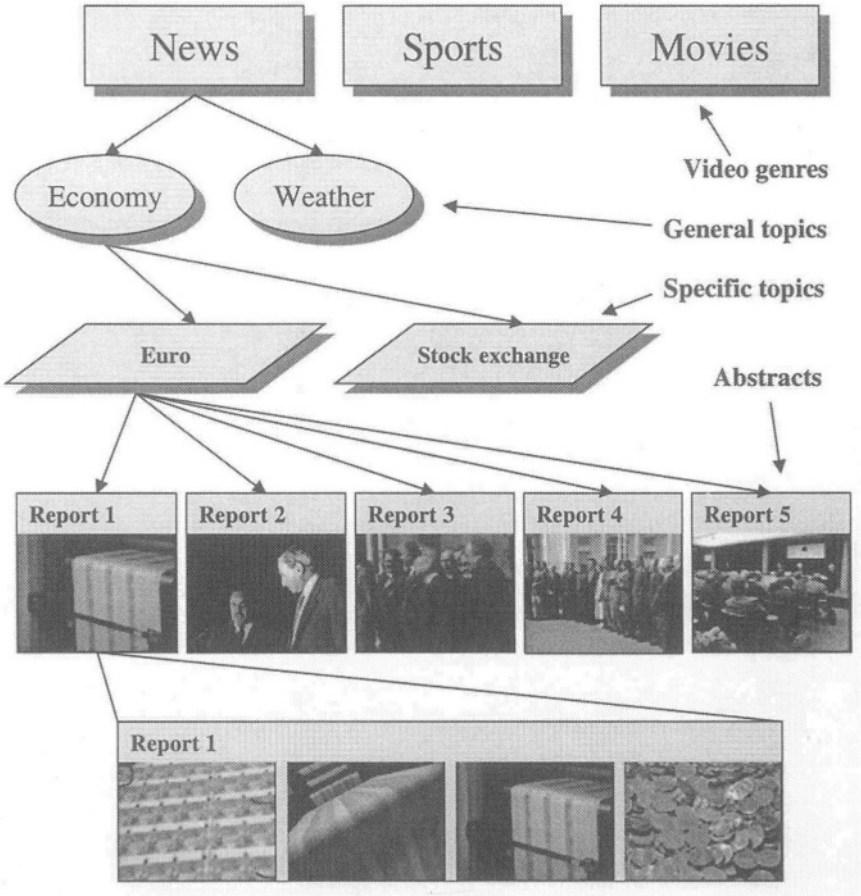

Figure 4-3. An illustration of a hybrid video browsing interface. While index terms (labels) are used for topic preselection in the first interaction steps, (audio)visual abstracts may be employed at lower levels of the content representation hierarchy to speed up and ease the interaction.

A better alternative for organizing interaction with a large video collection at different content resolutions may be to combine index terms with (audio)visual *abstracts* of temporal video segments. For instance, while a first selection of potentially interesting video clips can be obtained on the basis of index terms, further selection steps can be based on looking at the content of video abstracts. There are several advantages of this approach compared to the approach based on using index terms only. First, as "an

image is worth more than a thousand words", the size of the hierarchical tree can be reduced considerably by using (audio)visual information at lower tree levels. In the example illustrated in Figure 4-3, video abstracts generated as collections of representative video frames (keyframes) are used to replace entire subtrees of terms that may be necessary to distinguish and describe in detail all reports in the video collection linked to the label "Euro". In this way the number of steps leading to the video clip of interest may be reduced considerably. Second, a hybrid interface of this kind is more suitable to serve different users, also those not being involved in the label definition process. Namely, while the high-level semantic concepts represented by index terms at the top of the tree (e.g. video genres, general and specific topics found within these genres, as indicated in Figure 4-3) are meaningful and relevant for a broad range of different users, the subjectivity of the indexing increases with each further tree level. In the extreme case, the index terms at the bottom of the tree may only be informative and meaningful for the person who defined them. However, if the (audio)visual video abstracts are used at lower tree levels instead, they can unify many different search requests in a single representation. Finally, video abstracts can be generated automatically without any user input.

In this chapter we provide an insight into the principles and possibilities for developing video indexing and abstraction techniques, which can enable an efficient and effective interaction with a large video collection.

4.2 VIDEO INDEXING

In general, the problem of video indexing can be seen as a pattern classification problem. In this sense, an automated video indexing algorithm could be developed that assigns a label to a video clip according to the features (properties) of the data found in the clip. These properties jointly form a *pattern* that is used by the algorithm to investigate the presence of a link between the data and the content described by the label.

A rather simple but illustrative example of searching for patterns in video data for the purpose of video indexing is the early approach of Yeung and Yeo [Yeu97a-b]. There, the results of *time-constrained clustering* of video shots (see Chapter 3) serves as the basis for classifying the series of shots as *dialog* or *action,* events using heuristic rules.

In the first step, all shots belonging to the same cluster are assigned one and the same cluster label. Then, by replacing each shot of a video by its corresponding cluster label, the entire video can be represented as a series of labels, that is,

$$A\,B\,C\,A\,D\,G\,H\,B\,A\,C\,D\,K\,H\,D\,B\,A\,C\,\ldots \qquad (4.1)$$

In the second step, the label sequence (4.1) is examined for the appearance of patterns that correspond to dialogs or actions. Yeung and Yeo use the term "dialog" to describe "an event of actual conversation or a conversation-like montage of two or more concurrent processes" [Yeu97b]. Such events mostly consist of recurring shots showing either the parties involved in a conversation or the segments of concurrent processes, possibly interspersed by a number of "noise" shots (e.g. the establishing shot or the shots of other parties not participating in a conversation). The pattern indicating a dialog event is detected using the following set of rules:

- a maximally long pattern of alternating cluster-labels is found,
- each of the most dominant cluster labels occurs at least twice,
- the number of "noise" cluster labels does not exceed a prespecified maximum.

Two patterns revealing the presence of the dialog event are indicated in bold in the example shot sequence (4.2) with the cluster-label *F* being the "noise" label.

$$X\,Y\,Z\,\mathbf{A\,B\,A\,B\,A\,B}\,C\,\mathbf{D\,E\,F\,E\,D\,E}\,G\,H\,I\,\ldots \tag{4.2}$$

Yeung and Yeo refer to an *action* event as a "progressive presentation of shots with contrasting visual contents to express the sense of fast movement and achieve strong emotional impacts" [Yeu97b]. The action segments are primarily used to rapidly unfold the story and are typically characterized by a highly dynamic scenery and camera work. In view of this, the rule set introduced by Yeung and Yeo to find action segments primarily searches for the longest possible patterns characterized by the minimal repetition of cluster labels: the boundaries of an action event are found when the number of distinct cluster labels appearing in a shot sequence becomes sufficiently close to the total number of shots in that sequence. In the example (4.3) an action segment is marked (in bold) consisting of 11 shots with 9 distinct shots. Recurrences of labels *B* and *E* serve to indicate that certain minimum repetition of the video content across an action segment is allowed.

$$\mathbf{A\,B\,C\,D\,E\,B\,F\,G\,E\,H\,I}\,B\,A\,B\,C\,E \tag{4.3}$$

In the video-indexing example described above the presence of a semantic concept (dialog or action) was inferred on the basis of measurements (clustering) performed on video shots and by using a suitable set of rules. The rule-based inference is only an example of many knowledge inference techniques that are known in the theory of pattern classification

and that may be used for the purpose of video indexing. Further, the information serving as input for the inference process described above is rather simple, as not more than the global similarity of the visual contents in consecutive video shots is used. For detecting dialogs and actions in a more robust way in a variety of videos, however, and for revealing more complex semantic concepts from data in general, sophisticated mechanisms for *content modeling* are required.

4.2.1 Content modeling

Given a particular video clip, its content can be indexed in various ways. Although the number of possible content aspects, for which labels could be defined, is infinite, most of these aspects fall into one of the three levels in the general hierarchy of semantic concepts, as illustrated in Figure 4-4.

The top of the hierarchy is characterized by "topics" describing the most general content aspects of a video clip. A topic is typically defined at the level of a semantic segment, like the episode in a movie or a story unit (report) in a broadcast news program. Examples of topic labels are the "weather" news report and an "action" movie scene. However, topics can also be defined for the segments extracted from non-parsable videos, such as "suspicious human behavior" in a surveillance video.

The middle level of hierarchy consists of "events". Events are narrower semantic concepts than topics. They evolve over time and are characterized by the dynamics of the audiovisual content, like for instance, object(s) transformation and motion, camera work and temporal audiovisual effects. Events can be seen as the components of a "topic" semantic concept. For instance, an "action" scene in a movie may be characterized by the events of "explosion", "car chase", "helicopter flying" and "gunshots". Similarly, the topic "suspicious human behavior" will typically consist of the events corresponding to specific gestures, human-body motion or audio effects like screaming or shouting. In some video genres, however, no meaningful topics can be defined. Then, the events can be seen as the highest semantic concepts for indexing in these genres. A good example of such a case is a soccer broadcast in which the hierarchy of content labels typically does not go higher than the events of, for instance, "goal", "corner kick", "free kick" or "red card".

The lowest level of the hierarchical structure in Figure 4-4 contains the "sites", that is, locations where "events" take place, and "objects" that participate in the events. Examples of site labels are "indoor", "outdoor", "cityscape", "landscape", "mountain" and "forest", while object labels can be thought of as "car", "helicopter", "building", "face" or "person A". Here, only the static site and object instances are considered. Any site or object

dynamics considered by the label, like object motion or site transformation (e.g. rising sun, erupting volcano) will transfer the label to the level of "events".

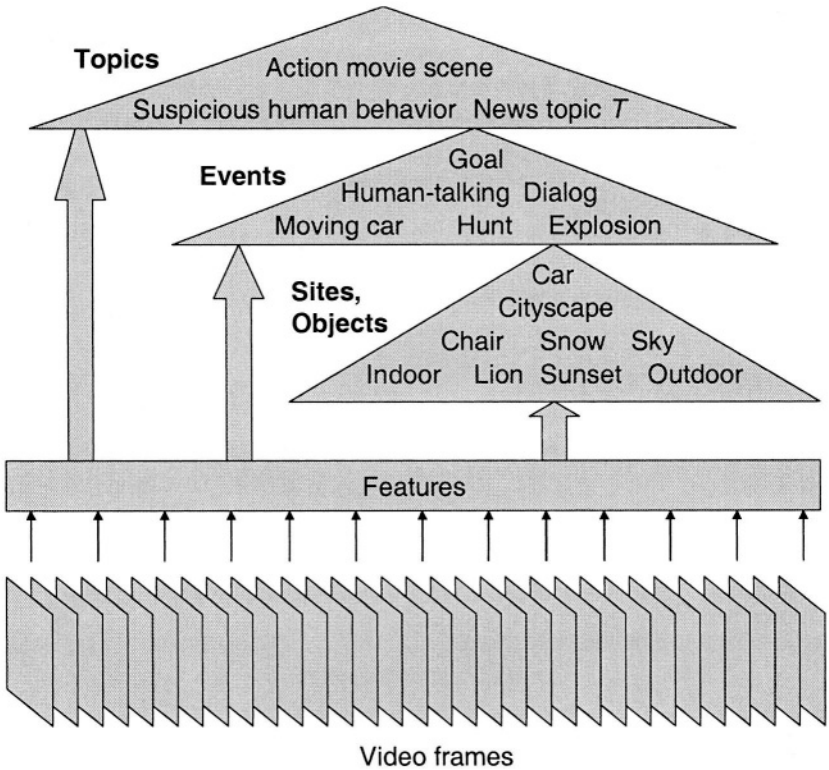

Figure 4-4. General hierarchy of semantic concepts

In order to be able to automatically assign a content label to a video clip, content models need to be developed that associate the features computed from data with semantic concepts. Then, the fit between a given video clip and the content model developed for the label *X* can be used as an indication of the presence of the semantic concept *X* in the clip and, consequently, as the criterion for assigning the label X to that clip.

The hierarchy in Figure 4-4 suggests a bottom-up approach to content modeling. The lower-level semantic concepts can be modeled first and then the obtained results can be aggregated into the models of the higher-level semantic concepts. In other words, "events" could be modeled on the basis

of the models of "sites" and "objects", and the "topic" models can be expressed as functions of the models of different "events". Due to a broad range of complexity of semantic concepts within each level of hierarchy, this bottom-up modeling approach is also applicable to the semantic concepts of the same kind. For instance, the low-level concept "Outdoor" can contain a number of less complex semantic concepts belonging to the same level of hierarchy in Figure 4-4, such as "Sky", "Snow" or "Water". In the same way, we can distinguish between more and less complex "events", like, for instance "Car chase" compared to "Moving car", or "Hunt" compared to "Fast moving animal".

For modeling semantic concepts at different levels of hierarchy the theory and tools of pattern classification can be applied. While modeling the "static" concepts of sites and objects should be approached using static pattern recognition techniques, "events" require the tools for time-series classification that optimally capture their content dynamics. Naphade and Huang [Nap02] argue that modeling of "objects" is much easier at the "event" level. In other words, instead of trying to model a static object, which requires a robust segmentation of static object shape from the static background, we can model the object as an integrated part of an event in which it participates. For instance, it is much easier to model a flying airplane than it is to model a static airplane, not only because of the additional information on object (airplane) motion relative to the background but also due to the possibility to use the information contained in the accompanying audio track (if available).

4.2.1.1 Modeling low-level semantic concepts

We illustrate the possibility for modeling semantic concepts on the idea of Naphade et al. [Nap98] that is based on generic probabilistic multimedia objects – *multijects,* A multiject can be seen as a system giving as output the probability that the semantic concept represented by the multiject is present in a video clip, given the features computed in the clip and the probabilities of the presence of other semantic concepts in that clip. The input from multijects that correspond to other semantic concepts is weighted according to *a priori* correlation between the semantic concepts. In the example in Figure 4-5, the detection of the concepts of "Sky" and "Snow" reduces the probability for the detection of the concept "Indoor" as "Sky" and "Snow" are not likely to be found in the video clip taken on an "Indoor" location. Similarly, the concept "Indoor" becomes more probable if a "Chair" or a "Bed" were already found in the clip. Consequently, the weights of the inputs coming from the multijects "Sky" and "Snow" will be strong but negative, indicating that these multijects are strongly anti-correlated with the

multiject "Indoor". In contrast to this, strong positive weights of the inputs from the multijects "Chair" and "Bed" enhance the support of these multijects for a more reliable detection of the "Indoor" concept.

Naphade et al. model the multijects of the low-level semantic concepts using Gaussian mixture models [Dud01], the parameters of which are estimated on the basis of the expectation-maximization (EM) algorithm [Dem77] and an adequate training data set.

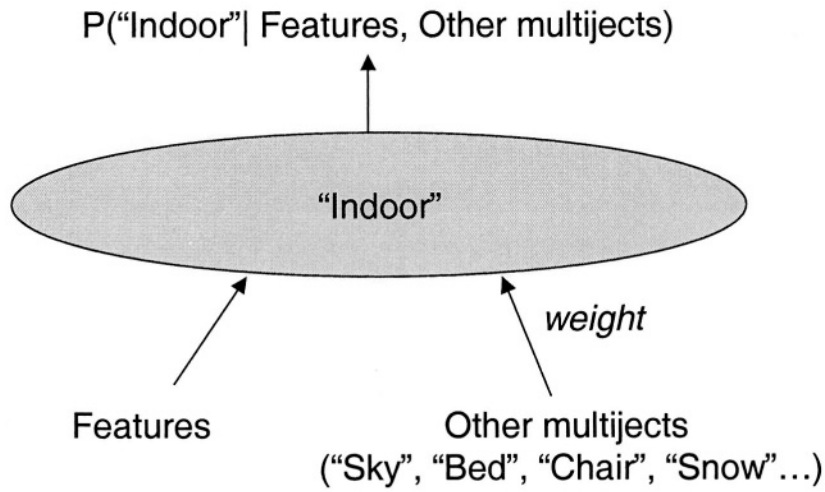

Figure 4-5. General structure of the "Indoor" multiject

4.2.1.2 Modeling medium-level semantic concepts

We saw earlier in this chapter that rather straightforward but effective rule-based models can be designed for the purpose of event detection. Figure 4-6 shows the state-diagram idea of Haering et al. [Hae00] to model the "hunt" events in a nature documentary. The state diagram is generated using the rules derived from observation and experimentation with a number of nature documentaries. The rules are based on the assumption that a hunt typically consists of a series of shots featuring smooth but fast animal motion, which is followed by the shots showing slower or no animal motion. In this sense, the detection of the "hunt" event is based on the detection of a less complex event that we could label as "Fast moving animal". The shot in

which a "Fast moving animal" is detected for the first time is marked as the starting shot of a potential "hunt" event. With this shot the event detection process goes from the default "Non-hunt" state to the "Beginning of Hunt" state. In order to confirm the instance of the "Hunt" event, the concept of "Fast moving animal" needs to be found in three consecutive shots. Only then the state of "Valid hunt" is reached and the "Hunt" event is recognized as such. The first subsequent shot in which the above moving object is not found marks the "End of hunt" state, which is eventually followed by the default state.

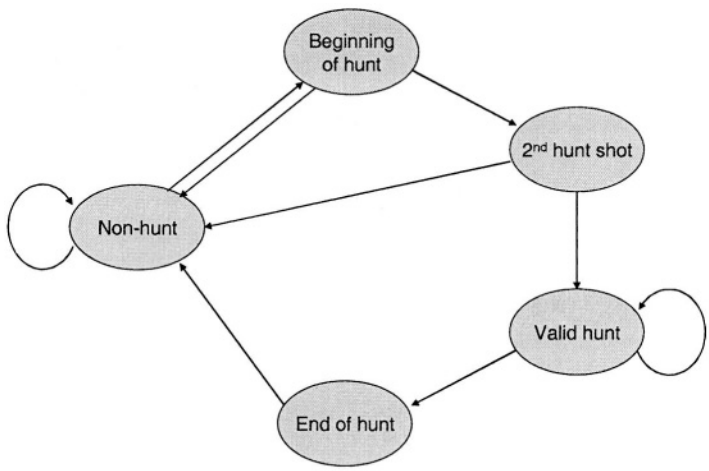

Figure 4-6. State diagram of the rule-based "Hunt" detector [Hae00]

The state-based event model described above can also be developed using the theory of hidden Markov models [Dud01]. Hidden Markov models (HMM) have proven to be among the most practical and effective mechanisms for modeling time-varying patterns. The HMM-based models are particularly powerful in representing the events that are characterized by a specific temporal pattern of the appearance and behavior of features and low-level semantic concepts across subsequent temporal video segments. A good example of such an event is a "dialog". Ferman and Tekalp [Fer99] proposed a hidden Markov model of a "dialog" event as illustrated in Figure 4-7. There, the states "Est" and "Master" correspond to the *establishing* and *master* shots, respectively. The establishing shot serves for setting up the location for the following action. The master shot provides the view of all

the characters in the scene. The states "1-shot" and "2-shots" refer to shots that contain 1 or two persons, respectively. Although the model may be extended by the states representing shots where more than two persons appear ("3-shot", "4-shot", etc.) the shots containing one or two persons are most common for a dialog sequence. All other cases are represented by the miscellaneous ("misc") state. Possible paths through the model are indicated by the arrows, where each arrow is characterized by the conditional probability of the corresponding HMM-state transition. Formally, we can define the HMM model for a "dialog" event as

$$\lambda_{dialog} = \left(\pi_{dialog}, \boldsymbol{A}_{dialog}, \boldsymbol{B}_{dialog}\right) \tag{4.4}$$

with π_{dialog} being the vector of the initial state probabilities, and with $\boldsymbol{A}_{dialog}$ and $\boldsymbol{B}_{dialog}$ being the state transition and confusion matrix, respectively. The elements of the matrices in (4.4) can be estimated from the training data set using, for instance, the Baum-Welch algorithm [Dud01]. Then, using the model (4.4) the likelihood $P(O|\lambda_{dialog})$ of the optimal state sequence O of the model can be computed, on the basis of which the posterior probability of a dialog, given the observations and priors, can be obtained. This posterior probability can be seen as an evaluation of the fit between the semantic concept "dialog" and the content of the video clip, and thus as the criterion for assigning the label "dialog" to that clip.

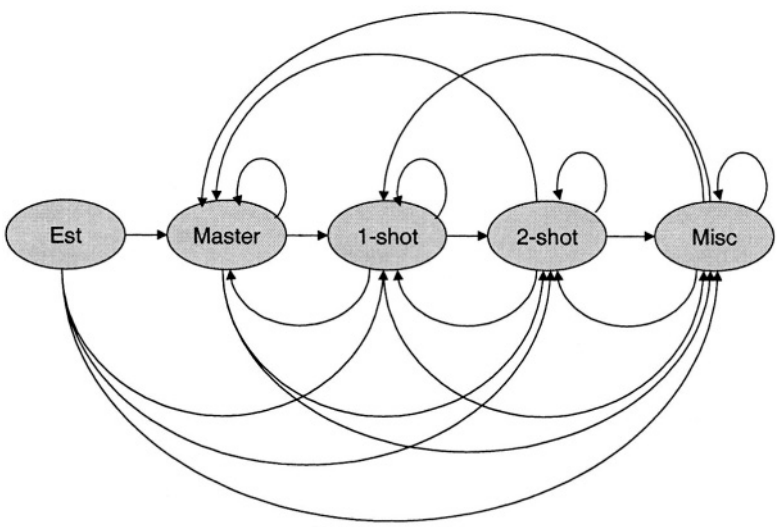

Figure 4-7. A hidden Markov model for dialogs [Fer99]

Clearly, an HMM-based event model can be represented in the same way as the multiject in Figure 4-5, having as inputs the features computed in video (serving as observations for the HMM) and outputs of other multijects (contributing to the prior probability of the event), while providing at its output the posterior probability of the event being modeled. Therefore, HMM-based event models provide means for extending the multiject-based semantics modeling approach from low-level to medium-level semantic concepts [Nap98].

For modeling complex events, the reliable detection of which strongly depends on optimally fusing the information contained in different modalities of video, a number of extensions of the basic HMM principle can be used. Examples of such extensions are *event-coupled HMM* and *hierarchical HMM* [Nap98]. For instance, in the hierarchical HMM, first, a separate component-HMM can be built and applied to each of the modalities. Then, the optimal state sequences of the component-HMMs can serve as observations for the supervisor-HMM that evaluates the correlation of these sequences and, finally, emits the probability of event occurrence.

4.2.1.3 Modeling high-level semantic concepts

As we already discussed in Section 4.2.1.1, a multiject can benefit from the outputs of other multijects to enhance the detection of the semantic concept it represents. To maximize this benefit, the process of bringing the multijects in relation to each other can be continued until a large number of multijects are connected into a network – a *multinet* [Nap98] – with multijects as nodes and with edges representing the interactions between the corresponding semantic concepts. The interactions between the multijects take into account the domain (prior) knowledge that specifies the likelihood for cooccurrence of different semantic concepts, but also various other contextual constraints like, for instance, the spatio-temporal ones. Examples of the latter are the hard constraint that "Sky" is always above "Water", and the event-based constraint that for detecting the event of "Human talking" the speech segment must be synchronous with facial expressions. Domain knowledge is embedded in the multinet via a set of rules or through the specification of prior probabilities. In the example of the multinet illustrated in Figure 4-8, the domain knowledge regarding the concept cooccurrence is represented by the positive and negative signs characterizing some of the graph edges. For instance, while the cooccurrence of the concepts "shark" and "bird" is highly unlikely, the detection of a "shark" can be enhanced by knowing that "water" has already been detected.

Besides of providing the means to improve the detection of semantic concepts, the multinet can also enable the detection of complex semantic

concepts that are difficult to model independently. For example, the complex "beach" concept can be inferred on the basis of the detection of multijects representing simpler concepts, such as "water", "sand" and "tree". At the same time, false detection of the "beach" concept can be prevented if the concept "indoor" has been detected, as these two concepts share a negative relation in the multinet. As the difficulty in semantics modeling increases with each higher level of the hierarchy in Figure 4-4, multinets appear to be a practical and effective tool for inferring the high-level semantic concepts. Getting back to the examples from the introduction to the Section 4.2.1, a multinet can be applied to detect an "action" movie scene on the basis of the presence of the events such as "explosion", "car chase" and "gunfire".

Naphade and Huang propose two ways of integrating domain knowledge regarding the dependencies between semantic concepts and the propagation of the impact of evidence on the probabilities of outcomes throughout the multinet, namely, by using Bayesian belief networks [Nap01, Dud01] and factor graphs [Nap02, Fre98, Ksc01].

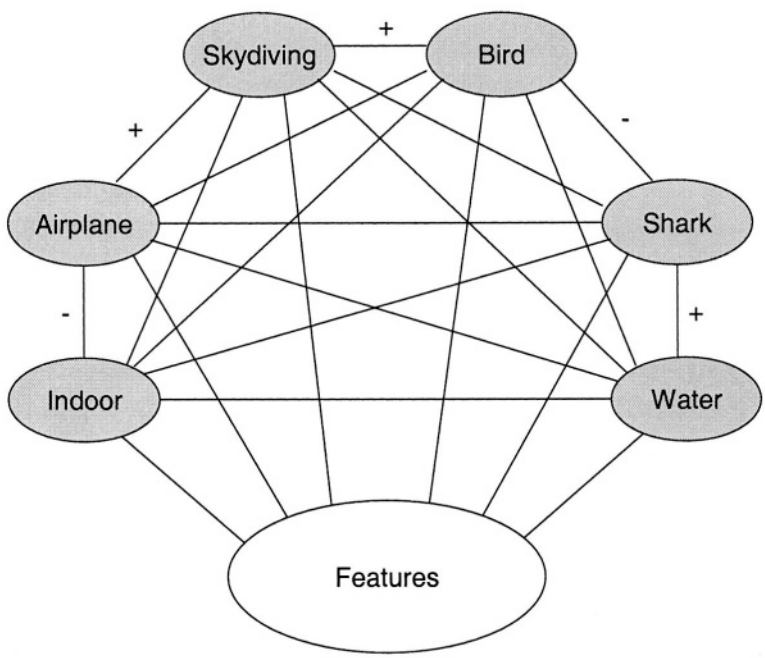

Figure 4-8. An illustration of a multinet.

4.2.2 A different example: News indexing

The problem of video indexing cannot always be approached through content modeling in the way described in the previous section. From the discussion in Chapter 3 we learned that in television news programs and similar video genres the audiovisual features extracted from a video clip are not capable of providing a reliable base for revealing the content of that clip. Consequently, detecting the concepts of "sites", "objects" and "events" in a news video is not likely to lead to a successful detection of the reports on given topics. Clearly, alternative techniques to those described above are required to reliably index a news television broadcast. We demonstrate the possibilities for developing such techniques on the example of the ***D**elft **A**dva**NC**ed n**E**ws **R**etrieval **S**ystem* (DANCERS) [Han01].

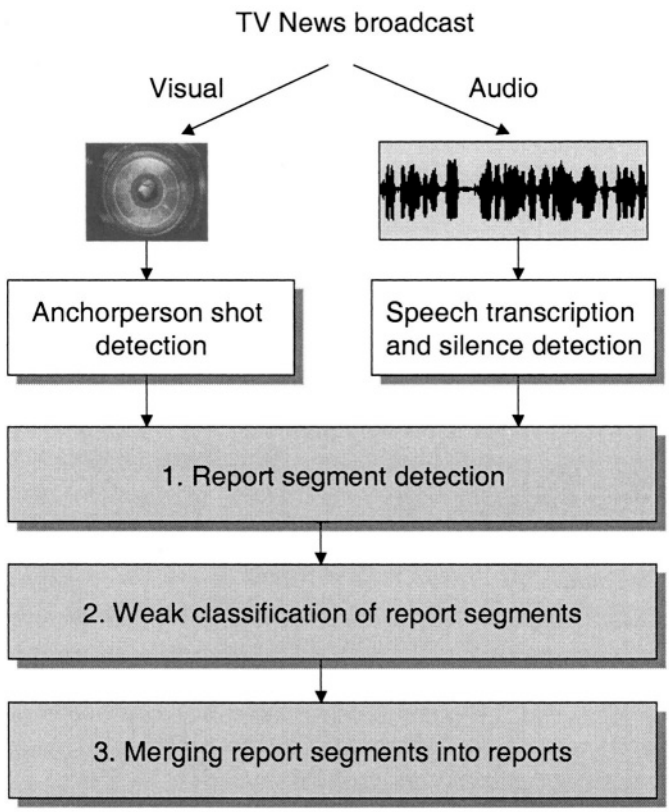

Figure 4-9. An overview of the DANCERS news-indexing scheme [Han01]

The objective of the DANCERS is to recognize the temporal segments in a news television broadcast that cover the topics from the list specified a priori by the user. As indicated in Figure 4-9, the DANCERS operates in three global steps:

- Report segments detection
- Weak classification of report segments
- Merging report segments into reports

In the first step a news program is partitioned into *report segments.* This is a multimodal processing step as the information from both the visual and audio track of video is used. The report segments are defined such that their boundaries are the potential report boundaries. Consequently, the reports will be found by merging the related neighboring report segments. One set of boundaries of report segments is found at places of silences in the audio track. This is based on the assumption that the anchorperson is likely to briefly pause in reading after completing one story and before introducing the following one. Another set of report segment boundaries is obtained by collecting the beginning and ending time stamps of anchorperson shots. Considering this set of time stamps is justified by the assumed relation between the editing structure of the news program (anchorperson shots interspersed with report shots) and the pattern of topic changes along the program. A variety of algorithms for anchorperson shot detection in video can be used for this purpose (e.g. [Ari96, Fur95, Han98]).

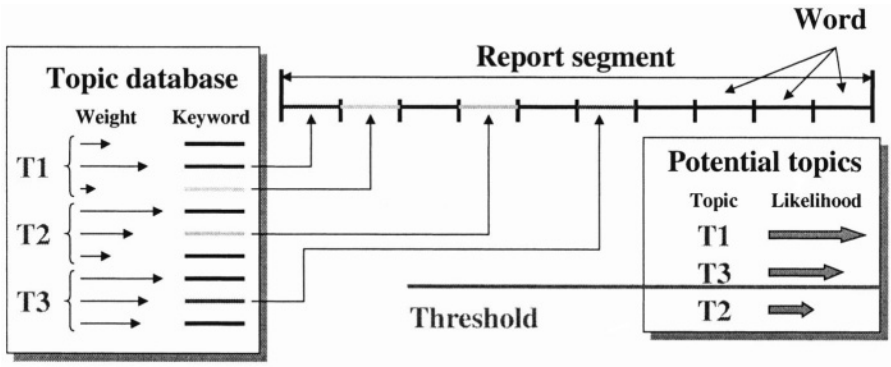

Figure 4-10. Assignment of the most probable topics per report segment

Once the report segments are found, each of them is assigned a list of topics that are most probably covered by that segment. For assigning a topic to a report segment reliable topic-specific keys are needed. We learned before that keywords extracted from the text of transcribed speech of the news broadcast are best capable of revealing the semantic content structure of the broadcast. For the purpose of keyword matching a database of topics is created prior to the news analysis process. For each topic being of interest to the user a vast number of keywords are collected. These keyword sets are typically generated by automatically inserting large text volumes related to the topics of interest, for instance, from the related Internet news sites. Each keyword in the database is also assigned a weighting factor (or weight) to quantify the importance of that keyword for a certain topic. We refer to the discussion in Section 3.4.2 regarding the computation of suitable weights for the keywords in text-based content analysis systems.

The scheme of the DANCERS module for topic-assignment per report segment is illustrated in Figure 4-10. The assignment starts with filtering the text of report segments and extracting only the words that are present in the topic database. Assuming that n different keywords from the database field related to the topic T_i are found in the report segment r, then the following set of rules is applied to determine the relevance and the likelihood $L_r(T_i)$ of topic T_i in segment r:

- Topic T_i is relevant for segment r if $n \geq c$. Here c is the critical number of different keywords from the keyword list of the topic T_i in the database, which need to be found in a segment in order to make the topic T_i relevant for that segment.

- If T_i is not relevant for the segment r, then the likelihood $L_r(T_i)$ is set equal to 0.

- If T_i is relevant for the segment r, then the likelihood $L_r(T_i)$ is computed as

$$L_r(T_i) = \frac{1}{l} \sum_{\substack{\text{sum over } l \text{ keywords } k \text{ with} \\ \text{highest weighting factors}}} w_r(k) \tag{4.5}$$

The formula (4.5) says that the likelihood that a report segment r covers the topic T depends on how unique and relevant the keywords found in that segment are for the topic T. The more unique and relevant the keywords, the higher the likelihood. The importance of the weighting factors $w_r(k)$ becomes obvious if we realize that only l keywords found for the topic T are

taken into account, that is, those having the highest weights. While the value of the parameter l cannot be smaller than c, we set an upper limit v to this value in order to prevent taking into account less relevant keywords that may confuse the subsequent segment merging process. The value for l is now determined using the following rule:

$$l = \begin{cases} c, & if\ n = c \\ n, & if\ c < n \le v \\ v, & if\ n > v \end{cases} \tag{4.6}$$

We compute the likelihood (4.5) for all relevant topics found in the segment r, and obtain a list of likelihood values, which can be sorted in descending order. This is also illustrated in Figure 4-10. The ordered likelihood value list can be defined as

$$\left\{\overline{L}_r(j) \,|\, \overline{L}_r(j) \ge \overline{L}_r(j+1),\ j = 1,..,\ p-1\right\} \tag{4.7}$$

with p being the number of relevant topics obtained for the segment r. In the next step we introduce thresholding of the ordered likelihood list (4.7) in order to separate the most probable from least probable topics per segment. The threshold value for segment r is determined as the likelihood value at which the largest "jump" in the ordered list (4.7) takes place, that is,

$$\begin{aligned} &L_{r,Threshold} = \overline{L}_r(j_{\max}+1) \\ &\text{where} \quad \left(\overline{L}_r(j_{\max}) - \overline{L}_r(j_{\max}+1)\right) = \max_{j=1,..,p-1}\left(\overline{L}_r(j) - \overline{L}_r(j+1)\right) \end{aligned} \tag{4.8}$$

Applying the threshold (4.8) splits the ordered list (4.7) into the list of the most probable and least probable topics. Only the most probable topics and their likelihood values are considered in the subsequent step of report generation. We denote the set of most probable topics for segment r by

$$\{T_{r,mp} \,|\, \overline{L}_r(j) > L_{Threshold} \wedge \overline{L}_r(j) = L_r(T_{mp}))\} \tag{4.9}$$

The input into the report-generating procedure consists of the report segments and their most probable topics $T_{r,mp}$ defined in (4.9). In the first step we merge all consecutive segments that contain the topic T_i within the list of their most probable topics. Performing this operation along the entire program results in a set of blocks at the lowest level of the segment-merging pyramid for the topic T_i, as shown in Figure 4-11. Blocks at higher pyramid

levels are obtained by merging all report segments that belong to the blocks at lower levels and those between them. In this way, blocks may appear that contain segments with no topic T_i in their lists of most probable topics. This is, for instance, the case with block 3 (segment 2), block 4 (segment 4) and block 5 (segments 2 and 4) in Figure 4-11.

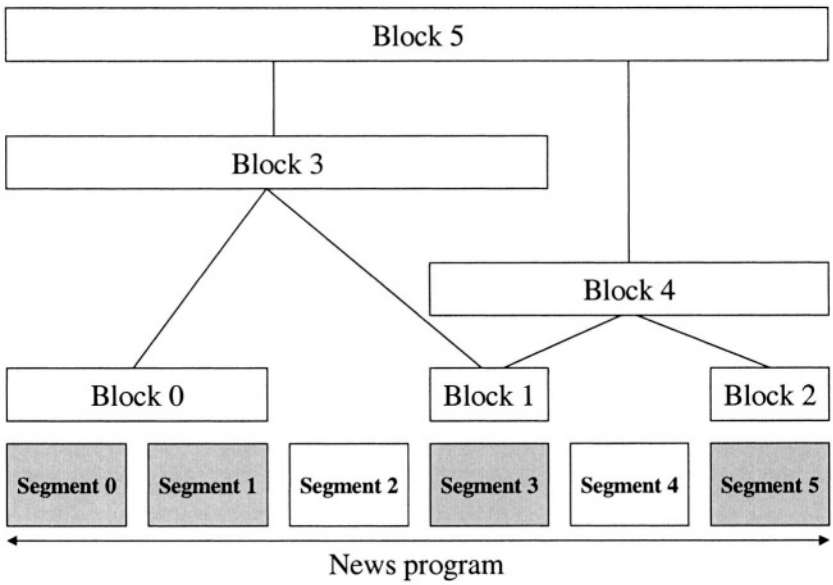

Figure 4-11. Illustration of the segment-merging pyramid for topic T_i and a news sequence consisting of six report segments. Segments characterized by grey blocks contain topic T_i in their lists of most probable topics

The report on topic T_i needs to be found among all blocks b_l (pyramid nodes) created for that topic. In the example shown in Figure 4-11, l has the values in the range of 0 to 5. Not all blocks b_l are, however, suitable to be considered as reports on topic T_i. The block b_l that belongs to the pyramid of the topic T_i is "valid", that is, it can be considered as a potential report on topic T_i, if it does not contain or is not contained by a block that belongs to the pyramid of another topic T_j and has a larger likelihood than the block b_l. This likelihood is computed as

$$L(b_l) = \frac{\sum_{q \text{ goes over all segments within the block}} L_q(T_i) N(q)}{N(b_l)} \tag{4.10}$$

The function $N(.)$ in (4.10) stands for the total number of words per segment (nominator) or per block (denominator). Then, the actual report on topic T_i is selected among all valid blocks for that topic as the block that

- is as large as possible,
- has the largest possible likelihood (4.10) (this requirement is considered only if two equally long valid blocks exist)

Performing the above procedure for the pyramids of all topics leads to the boundaries of news reports being of interest to the user.

We discuss the performance of the DANCERS on the example of a typical Dutch news broadcast lasting for 25 minutes and consisting of 11 reports on different topics. Report segments were generated using the time stamps of the anchorperson shots and of all silent segments lasting for more than 2 seconds. In this way 37 report segments were detected, numerated from 0 to 36. Then, for each segment the list (4.9) of most probable topics is created. For topic assignment per segment we created a database that covered 68 topics with in total 206 articles collected at various Internet news sites. The collection of articles contained about 7400 words. Topic assignment per segment was done using the values of the parameters c and v as 5 and 8, respectively. In total, 17 most probable topics were collected over all report segments. Finally, the pyramid (Figure 4-11) was created for each most probable topic. The total number of nodes in all pyramids was 45 and corresponds to the number of candidate news reports in our test sequence.

In Figure 4-12 we illustrate the composition of the pyramids on the example considering four topics. For these topics the pyramid nodes, their composition in terms of report segments, their likelihood and status (valid or not valid) are indicated. For instance, the pyramid for the topic "Euro" has 15 nodes, 10 of which are valid. Figure 4-12 also shows the results of the indexing process: the pyramid nodes indicated in bold are found as reports on the corresponding topics. A closer look at this process reveals that for the topic "Explosion" three blocks were found at the lowest pyramid level. After combining these initial blocks a total of 6 blocks were obtained. The blocks at higher pyramid levels have, however, low likelihood and are, therefore, overruled by the blocks of other pyramids having higher likelihood and containing the same report segments. For instance, Block 5 of the topic "Explosion" contains 32 report segments, has the likelihood of about 0.08 and is overruled by several blocks from other pyramids, like for instance, by the block 4 of the topic "Olympic games".

Of all nodes in Figure 4-12 only 3 nodes were selected as reports. It is important to note that, even if a pyramid consists of one block only (no other choice for that topic available), that block is not necessarily found as the report on that topic. A good example is the topic "Vietnam" in Figure 4-12.

The block representing the pyramid of this topic and consisting of one report segment has a relatively low likelihood, which makes it easily overruled by the blocks of other pyramids containing the same report segment but higher likelihood values.

Blocks per topic	Segments in the block	Likelihood	Status
Explosion			
0:	{3}	0.334558	valid
1:	**{17}**	**0.904958**	**valid**
2:	{35}	0.562923	valid
3:	{3,..,17}	0.176487	not valid
4:	{17,..,35}	0.124733	not valid
5:	{3,..,35}	0.081960	not valid
Olympic Games			
0:	{4}	0.287592	valid
1:	{27,..,29}	0.649827	valid
2:	{31}	0.869118	valid
3:	{4,..,29}	0.072887	not valid
4:	**{27,..,31}**	**0.656098**	**valid**
5:	{4,..,31}	0.112511	not valid
Euro			
0:	{7}	0.214011	valid
1:	{9,..,11}	0.862069	valid
2:	{13}	0.894411	valid
3:	{15,16}	0.259362	not valid
4:	{34}	0.588742	valid
5:	{7,..,11}	0.830162	valid
6:	{9,..,13}	0.872576	valid
7:	{13,..,16}	0.622709	valid
8:	{15,..,34}	0.039970	not valid
9:	{7,..,13}	0.855089	valid
10:	{9,..,16}	0.715388	valid
11:	{13,..,34}	0.160389	not valid
12:	**{7,..,16}**	**0.705103**	**valid**
13:	{9,..,34}	0.256822	not valid
14:	{7,..,34}	0.256245	not valid
Vietnam			
0:	{22}	0.183191	not valid

Figure 4-12. Examples of topic pyramids for some of the most probable topics detected in a test sequence. Bold blocks are selected as reports.

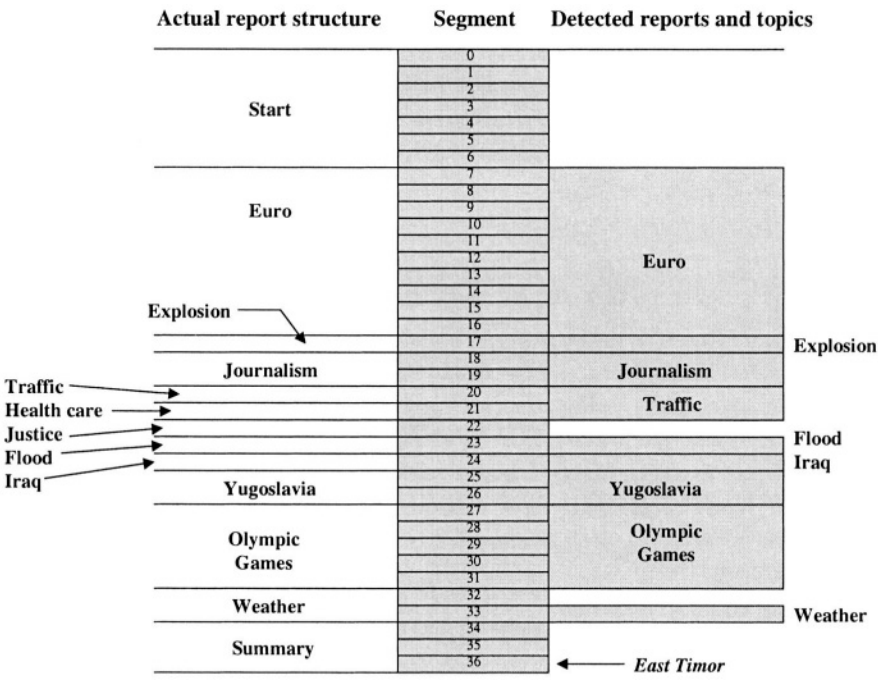

Figure 4-13. Actual temporal report structure of the test sequence and the indexing result: Properly detected reports and their topics (in boxes) and one false topic assignment.

Figure 4-13 shows the results of the segment merging process for the entire test sequence, compared to the real report structure of this sequence. The real structure of the sequence is shown on the left hand side while the reports found using DANCERS are listed on the right. In total, 10 reports were found, one of which ("East Timor") was false. Several important conclusions illustrating the performance of DANCERS can be drawn from the results presented in 4-13:

- If the topic database contains the topics that are present in the news program being analyzed, and if these topics are properly trained (e.g. using a sufficient number of news articles), then DANCERS is able to find the boundaries of reports corresponding to these topics. This can be seen on the example of topics "Euro", "Explosion", "Journalism", "Traffic", "Flood", "Iraq", "Yugoslavia", "Olympic Games" and "Weather".

- If the topic database contains a topic that is present in the news program being analyzed, but if not enough material was used to train the database for this topic, then DANCERS may fail in recognizing the corresponding reports. This was the case with the report on "Justice" in segment 22, for which we only used one short article to train the database.

- In order to be classified properly, a report segment has to be sufficiently long. If not, the classification result is unpredictable. A good example is the segment 36 consisting of several words only and therefore being falsely classified as a report on "East Timor".

- If a segment is too long, the probability increases that more than one topic is covered by that segment. Since only one topic per segment can be selected in the last instance, the indexing process may be disturbed in this case. For instance, the segment 21 covered both the topic "Health care" and the topic "Traffic", while only the last one was detected.

As can be seen in Figure 4-13, the segment-merging procedure provides an extraction of "relevant" news segments, that is, those complying with pre-specified topics, and neglects the other (irrelevant) parts of the program.

4.2.3 Multi-segment video indexing

The ideas described in previous sections can be applied to index a temporal video segment independent of the neighboring segments. Many parsable video genres are characterized, however, by the sequences of mutually related semantic segments. For instance, a typical TV news broadcast has a rather predictable structure. It may start by the reports on domestic politics, then continue with foreign politics, and finish - via the sport section - by the weather report. Clearly, the information on the content of one video segment provides in this case an additional clue that can be used to index the subsequent segments. For instance, in the news program structure mentioned above, it is highly unlikely that the segment following a report on domestic politics will be a weather report.

The relation between the contents of subsequent video segments can be exploited to enhance the indexing of each individual segment. As we again face the problem of classifying time-varying patterns, the same tools can be applied here as those already introduced in Section 4.2.1.2 for detecting "events" in video. Figure 4-14 illustrates the usage of a hidden Markov model for indexing an example sequence of six semantic segments.

Assuming that each segment can belong to N different semantic content categories (e.g. "topics"), these categories then determine N states of the model. In addition, the state N+1 is included to account for miscellaneous, otherwise unresolved entries. The model can be trained using the videos that have the "target" temporal content structure. The observation sequence of the model can consist of the feature patterns directly or can contain the results of content modeling per semantic segment. By evaluating the model on the given test sequence of semantic segments the most probable state sequence of the HMM is mapped to the segment sequence. This results in simultaneous indexing of all segments according to the optimal HMM state sequence.

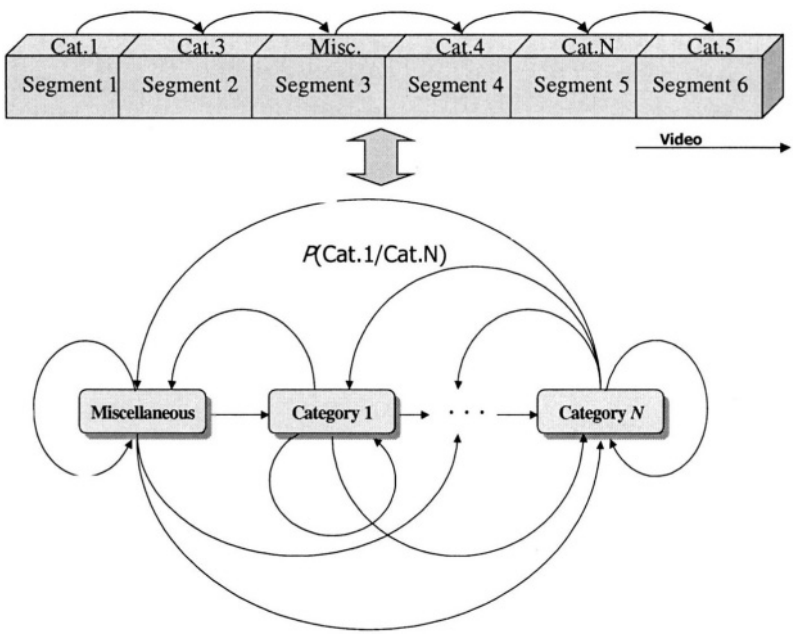

Figure 4-14. Multi-segment video indexing using hidden Markov models

4.3 VIDEO CONTENT REPRESENTATION FOR BROWSING AND CONTENT PREVIEW

As we discussed in the introduction to this chapter, efficient interaction with a large video collection at different content resolutions can be enabled by combining index terms with *abstracts* of video clips into hybrid video

browsing trees. A video abstract can be defined as a compact selection of the audiovisual material of a clip, representing the essence of the content of the clip. Since the richest information component of a video abstract is the visual one, video abstraction is sometimes also referred to as *video visualization.* Yeung and Yeo [Yeu97a] define video visualization as "the joint process of analyzing video and the subsequent derivation of representative visual presentation of the essence of the content".

We distinguish between two general types of video abstracts, namely the *static* and *dynamic* one. Dynamic abstracts consist of selected temporal parts of the original video clip. To generate such an abstract, prespecified classes of events can be extracted first from a video using the techniques discussed in Section 4.2, and then the event clips can be merged together in the resulting abstract video clip [Pfe96b, Bab02]. Clearly, the preselection of event classes to be included in a dynamic video abstract is highly genre dependent. The user study performed by Agnihotri et al. [Agn03] shows that, for instance, the abstracts of talk shows should show all guests and reveal all discussion topics, while the abstract of a news story should contain the people involved in the story, as well as the time and the location of the event covered by the story. The same study showed that movie abstracts should include the information on the characters and plot points but should not reveal the plot completely, as this would reduce the enjoyment of watching.

A dynamic video abstract can also be generated using the criteria that are not necessarily based on content modeling. A good example is the approach of Sundaram et al. [Sun02] who first define the measure of visual complexity of the analyzed video clip, and then compute and map this complexity onto the minimum length m of a frame sequence that is necessary to comprehend the content of the clip. Finally, a video abstract of the length m is generated using the procedure that maximizes the information content and the coherence of the abstract, given the constraints of multimedia synchronization, as well as the visual and audio syntax.

Another way of generating dynamic video abstracts is to use psychological criteria, such as visual attention [Ma02a]. To this class of approaches also belong those that are based on affective video content analysis. These approaches are treated in more detail in Chapter 5.

Still video abstracts consist of still images extracted from video or constructed from the selected frames of video. Due to their compactness, still video abstracts are particularly suitable for enabling quick browsing through the hierarchy of video content, and therefore for building hybrid hierarchical browsing trees, like the one sketched in Figure 4-3. However, dynamic video abstracts can be used in such a tree as well, for instance, to support a still image representing an event by a playable abstract of that event consisting of both the visual and the accompanying audio track.

The problem of creating still video abstracts was already addressed in Chapter 3 where we aimed at representing video clips in a compact fashion for efficiently computing their content similarity. There we mentioned the abstracts consisting of *keyframes* and *mosaics.* Keyframes and mosaics can also be applied in the context of content visualization for browsing. However, while mosaics can be generated here in the same way as described in Chapter 3, and employed directly for video browsing purposes, different criteria need to be used to extract keyframes and to organize them into an intuitive guide through the video content.

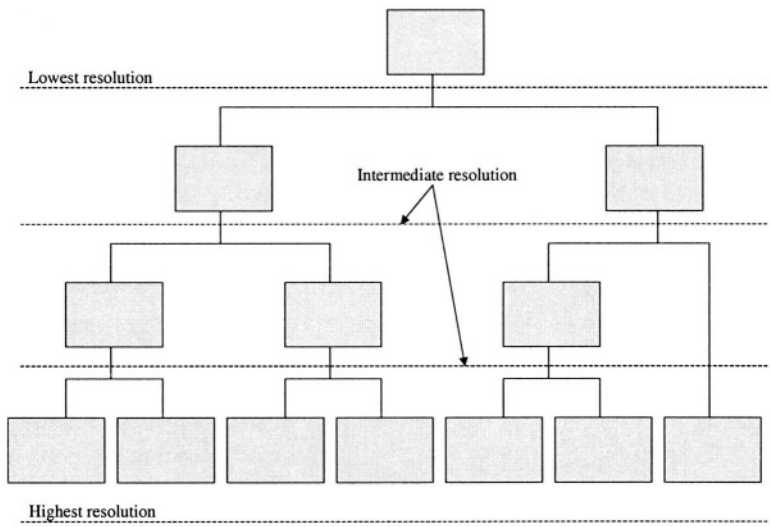

Figure 4-15. Keyframes organized using hierarchical clustering. Each level of hierarchy provides entry to the content at the corresponding resolution.

While keyframes extracted for video visualization purposes still need to capture all relevant aspects of the visual content of a video segment, and to minimize the redundancy in the visual content they capture [Han00], they also need to be meaningful to the user and provide sufficient information about the content of the segment they represent. In this sense, keyframes should be selected based on the criteria already discussed in Section 3.4.1.1, but also based on the importance of the persons and objects captured therein with respect to the overall content of the video segment. These extra criteria for keyframe selection would typically require additional processing steps, such as object or face detection and recognition. Further, of all the frames capturing the same content aspect, the most representative ones should be

extracted, like those taken under the best camera angle, or those collecting the maximum of important content elements (objects, faces, parts of the scenery). Finally, the keyframes extracted for the purpose of browsing and content preview should be technically acceptable as well: no blurred or dark frames should be extracted, neither the frames containing coding artifacts or interlacing effects.

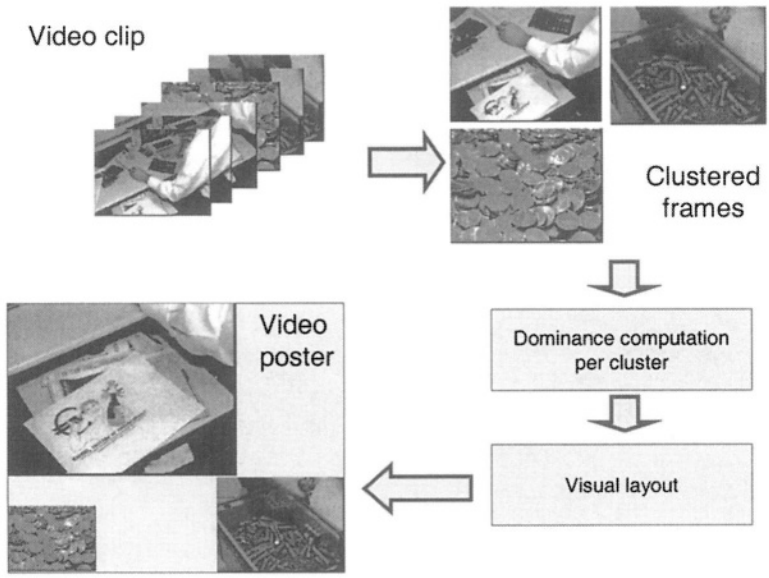

Figure 4-16. An illustration of the process of generating a video poster

Clearly, the task of automating the keyframe extraction process for video visualization purposes is not trivial and certainly marks one of the challenges to be met by future research in the area of video content analysis. This, however, should not prevent us to already think about suitable ways of organizing keyframes into the structures by which the user could be guided through the content of a large video collection in an efficient and effective manner. The simplest way of generating such structure is to apply a hierarchical clustering algorithm [Jai88] to the extracted keyframes. The result of such clustering is a tree, as illustrated by the example in Figure 4-15, that provides insight into the content of a video clip with increasing resolution at each deeper tree level [Oh00]. Naturally, in order for the tree to optimally lead the user through video content, the clustering criteria based on content semantics need to be applied. For defining these criteria, similar reasoning should be used as discussed above in the context of keyframe extraction.

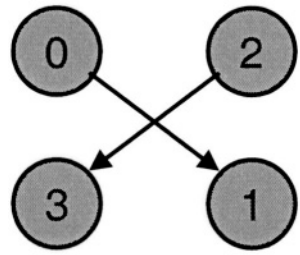

Figure 4-17. The underlying layout-principle for generating video posters

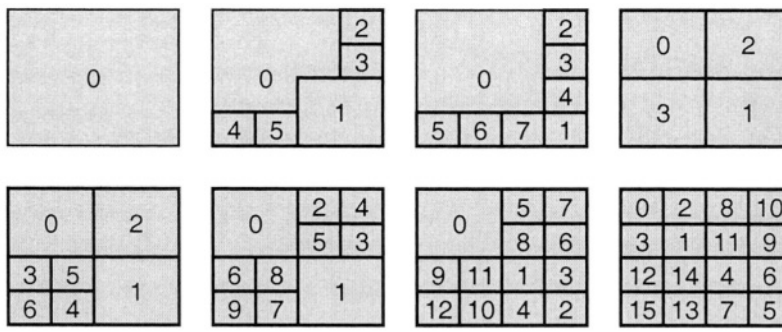

Figure 4-18. Examples of video poster layouts for different number of frame clusters

The concept of *video posters* proposed by Yeung and Yeo [Yeu97a-b], combines keyframes taken from different parts of the clip in the way to reflect the relative importance of the content of these clip parts in the overall content flow of the clip. The process of generating a video poster is illustrated in Figure 4-16. The process starts by grouping the frames of the clip into *N* clusters, each of which is then represented by one keyframe. Typically, the frame closest to the cluster centroid can be used for this purpose. The keyframe of each cluster is assigned a *dominance* value defining the relative importance of the content captured by the frames in the cluster compared to the contents of other clusters. The simplest way of measuring this importance is to look at the relative number of occurrences of the visual material of a cluster in the entire video clip. However, more elaborate content analysis techniques, for instance, those based on object and face recognition, could be developed and used for this purpose as well. In the last step, the keyframes are resized corresponding to their dominance

values and fitted within a selected layout pattern. Yeung and Yeo draw the rules for designing the poster-layout patterns from the practice of graphic design and newspaper editing and choose the design principle as illustrated in Figure 4-17. There, the numbers in the fields serve to indicate the importance of the field: the higher the number the lower the importance. Figure 4-18 shows the examples of the layout patterns generated using the abovementioned principle for different numbers of frame clusters.

4.4 REMARKS AND RECOMMENDATIONS

Although the research efforts aiming at the development of reliable video content modeling and abstraction mechanisms have been rather intensive in the past years, it would have been insufficient to provide in this chapter just an inventory of the existing approaches. This is mainly because the methods proposed so far are largely fragmented and aim at the indexing and abstraction solutions that are only usable either in controlled situations or in narrow application scopes. Instead, our objective in this chapter was to discuss the fundamental issues that should be addressed when developing reliable solutions to the video indexing and abstraction problems in a general case. In the specific context of video indexing, these issues include the following:

- Distinction between single- and multi-segment indexing,
- Suitability of particular features and content modeling approaches for the given indexing task,
- Selection of an effective way of integrating the available feature information, the dependences between the content elements found in video and other domain (prior) knowledge to maximize the quality of inference of the targeted semantic concepts.

The first item considers an important property of many video genres, that is, the relation between the consecutive temporal video segments. If present, this relation can be used as a valuable additional clue when indexing these segments. As we showed in Section 4.2.3, hidden Markov models can be used to generate a suitable indexing framework in this case.

The second item addresses the problem that is inherent in all video content analysis steps. However, this problem is particularly challenging in the context of video indexing due to a large variety of semantic concepts that may need to be modeled. Clearly, a useful step in simplifying this problem is to cluster semantic concepts into larger groups, like for instance, the three

levels of hierarchy in Figure 4-4, and to find ways to approach the modeling of semantic concepts in each of the groups in a unified way. When doing this, we can try to compensate for the incomplete or missing feature information by investigating the dependences between semantic concepts, as indicated in the third item in the list above. Although we illustrated the possibilities for modeling such dependences on the example of a unified probabilistic approach based on *multijects* and *multinets,* other tools of the pattern classification and knowledge inference theory can be applied for this purpose as well.

Regarding the problem of video abstraction, the biggest challenge is undoubtedly the extension of the current possibilities for generating dynamic and static video abstracts to the level where "meaningful" abstracts can be obtained, that is, the abstracts that are capable of "revealing the essence of the story" toward the user. Recent research results on generating dynamic video abstracts (e.g. visual skims) show a clear tendency in this direction. In contrast to this, the efforts on creating static video abstracts, especially keyframes, have been strongly biased by the ideas from the past, where the keyframe-extraction criteria were rather meaningless and formulated solely on the basis of low-level features, like, for instance, selecting keyframes at the points of minimum motion (object and camera activity) or by minimizing the redundancy in the visual content. While these "classical" approaches to keyframe extraction are still relevant to the processes of clip-to-clip comparison (see Chapter 3), the problem of keyframe extraction needs to be revisited when aiming at the browsing and retrieval applications.

The details of the example methods that we referred to in this chapter, but also of many inspiring ideas and methods dealing with various specific problems of video indexing and abstraction can be found in the literature below that we suggest for further reading.

4.5 REFERENCES AND FURTHER READING

[Agn03] Agnihotri, L.; Dimitrova, N.; Kender, J.; Zimmerman, J.: *Study on requirement specifications for personalized multimedia summarization,* Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Vol. 2 , pp. 757 – 760, 2003

[Ari96] Ariki Y., Saito Y.: *Extraction of TV news articles based on scene cut detection using DCT Clustering,* Proceedings of IEEE International Conference on Image Processing (ICIP), Vol. 3, pp. 847-850, 1996

[Bab02] Babaguchi N., Kawai Y., Kitahashi T.: *Event based indexing of broadcast sports video by intermodal collaboraton,* IEEE Transactions on Multimedia, Vol.4, No.1, March 2002

[Bim00] Del Bimbo A., Pala P., Tanganelli L.: *Retrieval by contents of commercials based on dynamics of color flows,* Proceedings of IEEE International Conference on Multimedia and EXPO (ICME), Vol.1, pp. 479-482, 2000

[Bra97] Brand M., Oliver N., Pentland A.: *Coupled hidden Markov models for complex action recognition,* Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 994-999, 1997

[Cha98] Chang S.-F., Chen W., Sundaram H.: *Semantic visual templates: Linking features to semantics,* Proceedings of IEEE International Conference on Image Processing (ICIP), Vol.3, pp. 531-535, 1998

[Che00] Chen W., Chang S.: *Generating semantic visual templates for video databases,* Proceedings of IEEE International Conference on Multimedia and EXPO (ICME), Vol.3, pp. 1337-1340, 2000

[Che02] Chen L., Ozsu, M.T.; *Rule-based scene extraction from video,* Proceedings of the IEEE International Conference on Image Processing (ICIP), Vol. 2 , pp. 737 – 740, vol.2, 2002

[Cla99] Clarkson B., Pentland A.: *Unsupervised clustering of ambulatory audio and video,* Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1999

[Dem77] Dempster A., Laird N., Rubin D.: *Maximum-likelihood from incomplete data via the EM algorithm,* Journal of Royal Statistical Society, Vol. 39, pp. 1-22, 1977

[Dim95a] Dimitrova N., Golshani F.: *Motion recovery for video content classification,* ACM Transactions on Information Systems, Vol. 14, No.4, pp 408-439, 1995

[Dim95b] Dimitrova N.: *The myth of semantic video retrieval,* ACM Computing Surveys (CSUR), Vol. 27, Issue 4, December 1995

[Dua03] Duan L.-Y., Xu M., Chua T.-S., Tian Q., Xu C.-S.: *A mid-level representation framework for semantic sports video analysis,* Proceedings of ACM Multimedia 2003

[Dud01] Duda R.O., Hart P.E., Stork D.G.: *Pattern Classification,* John Wiley & Sons, Inc. 2001

[Fer99] Ferman A.M., Tekalp A.M.: *Probabilistic analysis and extraction of video content,* Proceedings of IEEE International Conference on Image Processing (ICIP), 1999

[Fis95] Fischer S., Lienhart R., Effelsberg W.: *Automatic recognition of film genres,* Proceedings of ACM Multimedia, pp. 295-304, 1995

[Foo99] Foote J., Boreczky J., Wilcox L.: *Finding presentations in recorded meetings using audio and video features,* Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3029-3032, 1999

[Fre98] Frey B.J.: *Graphical models for machine learning and digital communication,* Cambridge, MA, MIT Press, 1998

[Fur95] Furht B., Smoliar S.W., Zhang H.: *Video and Image Processing in Multimedia Systems,* Kluwer Academic Publishers, 1995

[Gha97] Ghahramani Z., Jordan M.: *Factorial hidden Markov models,* Machine Learning, Vol.29, pp. 245-273, 1997

[Gir01] Girgensohn A., Boreczky J., Wilcox L.: *Keyframe-based user interfaces for digital video,* IEEE Computer, pp. 61-67, September 2001

[Hae00] Haering N., Qian R., Sezan M.I.: *A semantic event-detection approach and its application to detecting hunts in wildlife video,* IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No.6, September 2000

[Han98] Hanjalic A., Lagendijk R.L., Biemond J.: *Template-based detection of anchorperson shots in news programs,* IEEE International Conference on Image Processing (ICIP), Chicago (USA), 1998

[Han00] Hanjalic A., Langelaar G.C., van Roosmalen P.M.B., Lagendijk R.L., Biemond J.: *Image and video databases: restoration, watermarking and retrieval,* Elsevier Science BV, 2000

[Han01] Hanjalic A., Kakes G., Lagendijk R.L., Biemond J.: *Indexing and retrieval of TV broadcast news using DANCERS,* Journal of Electronic Imaging, 10(4), October 2001

[Iye98] Iyengar G., Lippman A.: *Models for automatic classification of video sequences,* Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases VI, pp. 216-227, 1998

[Jai88] Jain A.K., Dubes R.C.: *Algorithms for clustering data,* Prentice Hall, Advance Reference Series, 1988

[Kaw98] Kawashima T., Tateyama K., Iijima T., Aoki Y.: *Indexing of baseball telecast for content-based video retrieval,* Proceedings of IEEE International Conference on Image Processing (ICIP), pp. 871-875, 1998

[Kob00] Kobla V., DeMenthon D., Doermann D.: *Identifying sports video using replay, text and camera motion features,* Proceedings of SPIE Storage and Retrieval for Media databases, Vol. 3972, pp. 332-343, 2000

[Ksc01] Kschischang F., Frey B., Loeliger H.: *Factor graphs and the sum-product algorithm,* IEEE Transactions on Information Theory, Vol.47, pp. 498-519, 2001

[Li01] Li, B., Sezan, M.I.: *Event detection and summarization in sports video,* IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 132 – 138, 2001

[Li03a] Li B., Sezan M.I.: *Semantic sports video analysis: approaches and new applications,* IEEE International Conference on Image Processing (ICIP), 2003

[Li03b] Li D., Dimitrova N., Li M., Sethi I.: *Multimedia content processing through cross-modal association,* Proceedings of ACM Multimedia, 2003

[Liu98] Liu Z., Wang Y., Chen T.: *Audio feature extraction and analysis for scene segmentation and classification,* Journal of VLSI Signal Processing Systems, Special issue on multimedia signal processing Vol. 20, Issue 1-2 pp. 61-79, October 1998

[Ma02a] Ma Y.-F., Lu L., Zhang H.-J., Li M.: *A user attention model for video summarization* Proceedings of the ACM Multimedia, 2002

[Ma02b] Ma Y.-F., Lu L., Zhang H.-J.: *Motion pattern based video classification using support vector machines,* Proceedings of IEEE International Symposium on Circuits and Systems, Vol.2, pp. 69 – 72, 2002

[Nak97] Nakamura Y., Kanade T.: *Semantic analysis for video contents extraction: Spotting by association in news video,* Proceedings of ACM Multimedia, 1997

[Nap98] Naphade M., Kristjansson T., Frey B., Huang T.S.: *Probabilistic multimedia objects (multijects): A novel approach to indexing and retrieval in multimedia systems,* Proceedings of IEEE International Conference on Image Processing (ICIP), Vol.3, pp. 536-540, 1998

[Nap01] Naphade M., Huang T.S.: *A probabilistic framework for semantic video indexing, filtering and retrieval,* IEEE Transactions on Multimedia, Vol.3, No.1, March 2001

[Nap02] Naphade M., Huang T.S.: *Extracting semantics from audiovisual content: The final frontier in multimedia retrieval,* IEEE Transactions on Neural Networks, Vol. 13, No.4, July 2002

[Oh00] Oh J., Hua K.A.: *Efficient and cost-effective techniques for browsing and indexing large video databases,* Proceedings of the ACM SIGMOD international conference on Management of data, Volume 29 Issue 2, 2000

[Pan0l] Pan J.-Y., Faloutsos C.: *VideoGraph: A new tool for video mining and classification,* Proceedings of the first ACM/IEEE-CS joint conference on digital libraries, 2001

[Pfe96a] Pfeiffer S., Fischer S., Effelsberg W.: *Automatic audio content analysis,* Proceedings of ACM Multimedia, pp. 21-30, 1996

[Pfe96b] Pfeiffer S., Lienhart R., Fischer S., Effelsberg W.: *Abstracting digital movies automatically,* Journal of Visual Communication and Image Representation, Vol.7, No.4, pp. 345-353, December 1996

[Qia99] Qian R., Haering N., Sezan I.: *A computational approach to semantic event detection,* Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 1999

[Reh99] Rehg J., Murphy K., Fieguth P.: *Vision-based speaker detection using Bayesian networks,* Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 110-116, 1999

[Sau97] Saur D.D., Tan Y.P., Kulkarni S.R., Ramadge P.J.: *Automated analysis and annotation of basketball video,* Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases V, Vol. 3022, pp. 176-187, 1997

[Sud97] Sudhir G., Lee J.C.M., Jain A.K.: *Automatic classification of tennis video for high-level content-based retrieval,* Proceedings of IEEE International Workshop on Content-Based Access of Image and Video Database, pp. 81 – 90, 1998

[Sun02] Sundaram H., Xie L., Chang S.-F.: *A utility framework for the automatic generation of audio-visual skims,* Proceedings of ACM Multimedia, Juan Les Pins, France, December 2002

[Tov01] Tovinkere V., Qian R.: *Detecting semantic events in soccer games: Towards a complete solution,* Proceedings of IEEE International Conference on Multimedia and EXPO (ICME), 2001, pp. 833 - 836, 2001

[Vai98] Vailaya A., Jain A., Zhang H.: *On image classification: city images versus landscapes,* Pattern Recognition, Vol. 31, pp. 1921-1936, December 1998

[Vas97] Vasconcelos, N., Lippman, A.: *Towards semantically meaningful feature spaces for the characterization of video content,* Proceedings of IEEE International Conference on Image Processing (ICIP), Vol. 1, pp. 25 - 28, 1997

[Vas98] Vasconcelos N., Lippman A.: *A Bayesian framework for semantic content characterization,* Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 566-571, 1998

[Wac96] Wactlar H., Kanade T., Smith M.A., Stevens S.: *Intelligent access to digital video: The Informedia project,* IEEE Computer, Vol. 29, Issue 5, pp. 46 - 52 , May 1996

[Wan97] Wang Y., Huang J., Liu Z., Chen T.: *Multimedia content classification using motion and audio information,* Proceedings of IEEE International Conference on Circuits and Systems, pp. 1488-1491, 1997

[Wan00] Wang Y., Liu Z., Huang J.: *Multimedia content analysis using audio and visual information,* IEEE Signal Processing Magazine, Vol. 17, No.6, pp. 12-36, November 2000

[Who01] Ngo C.-W., Pong T.-C., Zhang H.-J.: *On clustering and retrieval of video shots,* Proceedings of ACM Multimedia 2001

[Wol96] Wold E., Blum T., Keislar D., Wheaton J.: *Content-based classification, search and retrieval of audio,* IEEE Multimedia, Vol.3, No.3, pp. 27-36, 1996

[Yeu97a] Yeung M., Yeo B.-L.: *Video visualization for compact presentation and fast browsing of pictorial content,* IEEE Transactions on Circuits and Systems for Video Technology, Vol.7, pp. 771-785, 1997

[Yeu97b] Yeung M., Yeo B.-L.: *Video content characterization and compaction for digital library applications,* Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases V, Vol. 3022, pp. 45-58, 1997

[Zha00] Zhang T., Kuo C.: *An integrated approach to multimodal media content analysis,* Proceedings of SPIE conference on Storage and Retrieval of Media databases, Vol. 3972, pp. 506-517, 2000

[Zho97] Zhong D., Chang S.-F.: *Spatio-temporal video search using the object-based video representation,* Proceedings of IEEE International Conference on Image Processing (ICIP), Vol.1, pp. 1-24, 1997

[Zho01] Zhong D., Chang S.-F.: *Structure analysis of sports video using domain models,* Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 713–716, 2001

[Zho00] Zhou W., Vellaikal A., Kuo C.-C. J.: *Rule-based video classification system for basketball video indexing,* Proceedings of ACM Multimedia, 2000

Chapter 5

AFFECTIVE VIDEO CONTENT ANALYSIS

5.1 INTRODUCTION

In this chapter we address the problem of extracting the *affective content* from video. The affective content of a given video clip can be seen as the amount and type of *affect* (feeling, emotion, mood) that characterizes that clip. As opposed to the *cognitive content* that we considered in this book so far and that is built of the facts about the temporal video content structure (content coherence), the objects captured by the camera, and the scene composition and type (like dialogs, actions, news or documentary topics), the affective content reaches beyond these facts. Assuming that a cognitive content analysis algorithm has been used to identify, for instance, all video clips showing dialogs, the affective analysis steps are required to re-filter the obtained clip set in order to identify those dialogs that are tense, relaxed, sad, or joyful. In this sense, affective video content analysis can be seen as an extension of the theory discussed in the previous chapter, by which the scope of content labels is broadened to capture not only cognitive but also affective semantic concepts.

Identifying the affective content of a given video clip is important for various video indexing and retrieval applications. To illustrate this we quote the statistical fact reported by Picard [Pic97a] that finding photographs having a particular mood was the most frequent request of advertising customers in a study of image retrieval made with Kodak Picture Exchange [Rom95]. One could easily extend this result to video collections as well: an average user will often search for the funniest or most thrilling fragments of a movie, as well as the most exciting segments of a sport event. On the other hand, the user may also wish to remove “unpleasant” video segments from

his video collection. Finally, as the user preferences are based to a great extent on the prevailing mood of a video, the affective video content analysis is therefore likely to provide valuable information that can be used in the process of personalizing the video delivery to the user.

The amount and type of affect characterizing a video clip are referred to in this chapter as those that are *expected* to arise in the user while watching the clip. This *expected* feeling or emotion can be seen as the one that is either intended to be communicated toward the audience (from video program directors), or that is likely to be elicited from the majority of the audience who are watching the particular video clip. To illustrate the former we use the quote of Ian Maitland [Pic97a] - the Emmy-Award-winning director and editor: *"It is the filmmaker's job to create moods in such a realistic manner that the audience will experience those same emotions enacted on the screen, and thus feel part of the experience.*" The expected affective response of a broad audience can best be illustrated by the example of a sport broadcast: A score (goal) in a soccer match can generally be considered a highly exciting event, just like the finish of a swimming competition or the sprint over the last 50 meters in a running contest.

At this stage it is worthwhile emphasizing that the affective content of a video does not necessarily correspond to the affective response of a particular user to this content. In other words, the *expected* feeling or emotion as described above should not be mixed up with the *actual* feeling or emotion that is evoked in a user while watching video. The expected affective response can be considered objective, as it results from the actions of the movie director, or reflects the more-or-less unanimous response of a general audience to a given stimulus. Opposed to this, the perceived feeling or emotion is highly subjective and context-dependent. Therefore, it may be very different from the expected one and may also vary from one individual to another. For instance, the same soccer television broadcast may make the winning team's fans happy, the losing fans sad, and elicit no emotions at all from an audience that is not interested in soccer. On the other hand, the relation between the expected and the subjective affective responses (e.g. marking a horror movie with the label "funny" for those people who always laugh while watching such movies) and the information about the context (e.g. winning or losing soccer fan) can be taken into account, for instance, by generating the profile of a particular user. This profile can be seen as a function mapping the expected affective response to a given stimulus onto the user-specific affective response to that stimulus. Once this function is known for a particular user, individual deviations between the elicited and expected mood can always be taken into account later on and used to "personalize" the expected mood accordingly.

5.2 DIMENSIONAL APPROACH TO AFFECT

A human affective response or state can be represented using the following three basic dimensions [Sch54, Osg57, Rus77, Bra94, Lan95a]:

- Valence (V)
- Arousal (A)
- Control (Dominance) (C)

Valence is typically characterized as a continuous range of affective responses or states extending from pleasant or "positive" to unpleasant or "negative" [Det97], while arousal is characterized by affective states ranging on a continuous scale from energized, excited and alert to calm, drowsy or peaceful. We can also say that arousal stands for the "intensity" of emotion, while valence can be related to the "type" of emotion. The third dimension – control (dominance) – is particularly useful in distinguishing among affective states having similar arousal and valence (e.g. differentiating between "grief" and "rage"), and typically ranges from "no control" to "full control". Consequently, the entire scope of human affective states can be represented as a set of points in the three-dimensional VAC coordinate system.

While we could tend to assume that the points corresponding to different affective states are equally likely to be found anywhere in the three-dimensional VAC coordinate system, psychophysiological experiments show that only certain areas of this system are relevant. These experiments typically include measurements of affective responses of a large group of subjects to calibrated audio-visual stimuli collected in the *International Affective Picture System* (IAPS, [Lan85]) and the *International Affective Digitized Sounds* system (IADS, [Bra91]). Subjects' affective responses to these stimuli can be quantified either by evaluating their self-reports, e.g. by using the Self-Assessment Manikin [Lan80], or by measuring physiological functions that are considered related to particular affect dimensions. For example, heart rate reliably indexes valence, where skin conductance is associated with arousal. It was found namely that the heart rate accelerates as a reaction to pleasant stimuli, while unpleasant stimuli cause the heart rate to slow down [Fit92, Gre89, Det97]. Also, an increase in arousal causes the sweat glands to become active and the skin conductance responses become larger and more frequent [Hop94, Det97]. While IAPS and IADS are specially created to evoke a wide range of different emotions with their audio-visual content, the three-dimensional surface circumventing the mappings of affective responses onto the 3D VAC coordinate system is roughly parabolic. An idea about the form of this surface can be obtained

from the illustration in Figure 5-1. The parabolic shape becomes logical if we realize that there are relatively few or even no stimuli that would cause an emotional state characterized by, for instance, high arousal and neutral valence, or high valence accompanied by low arousal [Die99].

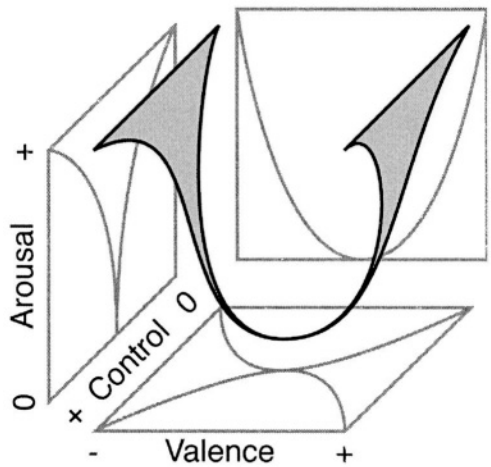

Figure 5-1. A view of the 3D affect space (adopted from Dietz and Lang [Die99])

The dimensional approach to representing affect, as described above, can play an important role in the development of agents that serve as mediators between the computer and the user, and involve the user in an interaction with the computer that closely resembles the interaction between humans [Nas94]. Since human-to-human interaction is strongly determined by emotions, the best agents are the "affective" ones, that is, those that are able to sense, synthesize and express emotions. For instance, Dietz and Lang [Die99] use the parabolic shape introduced above as the basis for assigning a temperament, mood and emotion to an affective agent and so for defining the "personality" of that agent. The temperament is a fixed point in the space that defines the "at rest" state of the agent (its rudimentary personality). While the temperament is static, the points corresponding to the mood and emotion of the agent can move freely within the space. The position of the emotion point gives rise to the expressions of the agent and determines its current affective state. Further, the emotion point gravitates toward the position of the mood that, again, moves through the space relatively slowly, is mainly pulled by emotional events and gravitates towards the position of the temperament. The dynamics of the system is therefore influenced by both the agent's current affective state and its temperament.

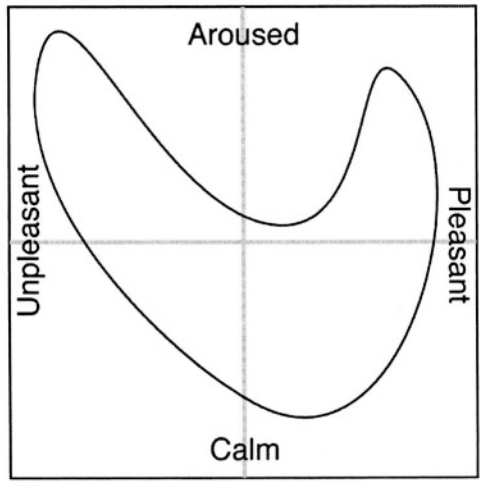

Figure 5-2. Illustration of the 2D affect space (adopted from Diets and Lang [Die99])

5.3 AFFECTIVE VIDEO CONTENT REPRESENTATION

5.3.1 2D affect space

As can be seen from Figure 5-1, the effect of the control dimension becomes visible only at points with distinctly high absolute valence values. This effect is also quite small, mainly due to a rather narrow range of values belonging to this dimension. Consequently, it can be said that the control dimension plays only a limited role in characterizing various emotional states. As a matter of fact, Greenwald et al. [Gre89] have shown that valence and arousal account for most of the independent variance in emotional responses. This is especially true for the problem addressed in this chapter - the extraction of the affective content from video. Numerous studies of human emotional responses to media have shown that "emotion elicited by pictures, television, radio, computers and sounds can be mapped onto an emotion space created by the arousal and valence axes" [Die99]. For this reason, we neglect the control dimension and consider the arousal and valence dimensions only. Instead of the three-dimensional surface introduced in the previous section, the relevant affect space for the purpose

of affective video content analysis is reduced to the projection of this surface onto the arousal-valence plane. Figure 5-2 shows an illustration of the resulting 2D affect space. The parabolic contour is generated to circumvent the scatter plot of affective responses with respect to arousal and valence only, which were collected using the IAPS and IADS stimuli. It is expected that the affective states extracted from a video can be represented as the points within this contour.

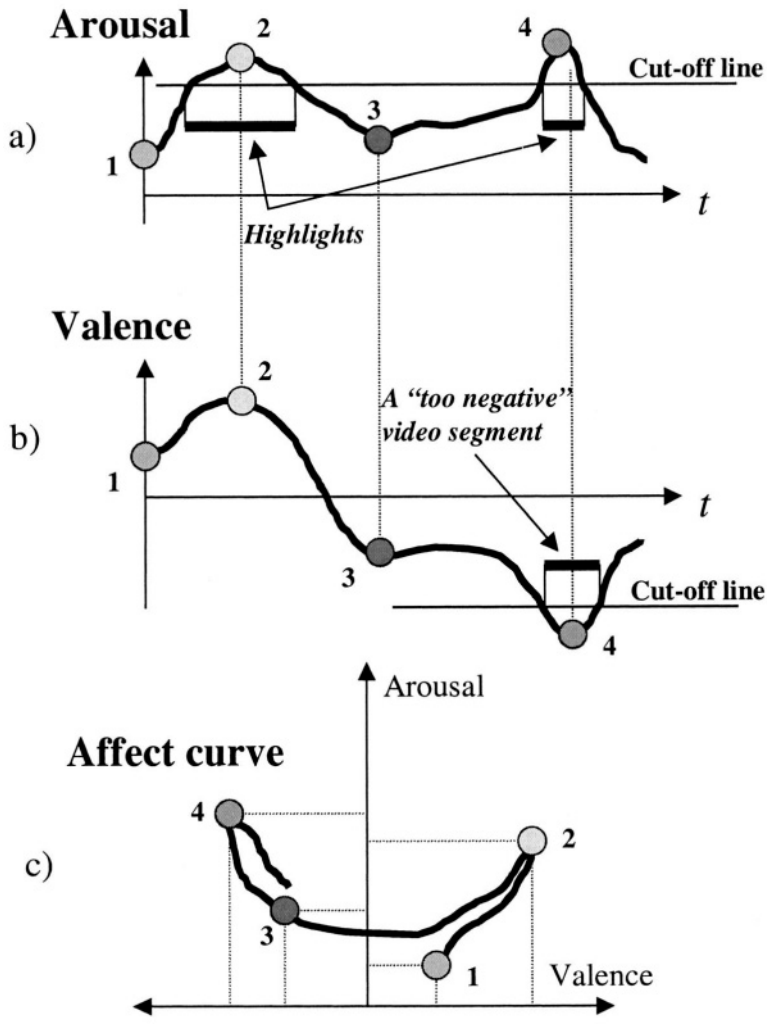

Figure 5-3. An illustration of arousal, valence and affect curve

5.3.2 Arousal, valence and affect curve

By computing the arousal and valence values along a video, a set of curves can be obtained that can provide a representation of the affective content of a video in view of the applications mentioned in the introduction to this chapter [Han01].

The *arousal time curve* indicates how the intensity of affective state changes along a video, and depicts the expected changes in user's excitement while watching that video. In this sense, the arousal curve is particularly suitable for locating the "exciting" video segments, and we will also refer to it later on as *excitement time curve*. On the basis of the arousal time curve we can generate a video abstract containing the highlights in a desired length. Namely, given the maximum abstract length N in frames, a horizontal line can be drawn cutting off the peaks of the curve in such a way that the number of frames covered by the peaks is not larger than N (Figure 5-3a).

The *valence time curve* depicts the state changes in the type of affective states contained in a video over the time. As such, this curve mimics the expected changes of "moods" of the user while watching a video. Using the valence time curve we can also determine the "positive" and "negative" video segments with respect to the expected type of feeling that is evoked in the user during these segments. This information can serve to match the video to personal preferences of the user, but also to automatically perform "censorship" tasks, that is, to remove all segments from a video that are "too negative" for certain groups of the audience. As illustrated in Figure 5-3b, such segments may be searched among those for which the valence curve reaches sufficiently low values.

The arousal and valence time curves can be combined into the *affect curve*. This curve is composed of the value pairs of the arousal and valence time curves that are taken per time stamp of the video and mapped onto the corresponding points of the 2D affect space (Figure 5-3c). The affect curve can be seen as the most complete representation of the affective content of a video, which can be obtained automatically. This curve can be interpreted in various ways and used for numerous applications related to video content representation and retrieval at the affective level. For instance, assuming that the affect curve has already been computed for a given video, an arbitrary temporal segment of that video can automatically be indexed with respect to the affective states through which the corresponding part of the affect curve passes. Indexes can be provided in the form of labels that are assigned a priori to different regions of the 2D affect space, as illustrated in Figure 5-4. Also, the area of the 2D affect space in which the curve traverses most of the time corresponds to the dominant affective state ("prevailing mood") of a

video. This can be highly useful for automatically classifying a video into different affective genres. Further, the affect curve may directly serve as a criterion for filtering the incoming videos according to a user's preference. This preference can, namely, be represented by the user profile consisting of affect curves of all programs that the user has selected in the past (in the learning phase of the system). Filtering an incoming video according to this preference is then nothing more than matching the properties of the affect curve of the incoming video with the properties of the affect curves included in the profile. We will address the possibility for generating and using a profile based on affect curves in more detail in Section 5.5.3.1.

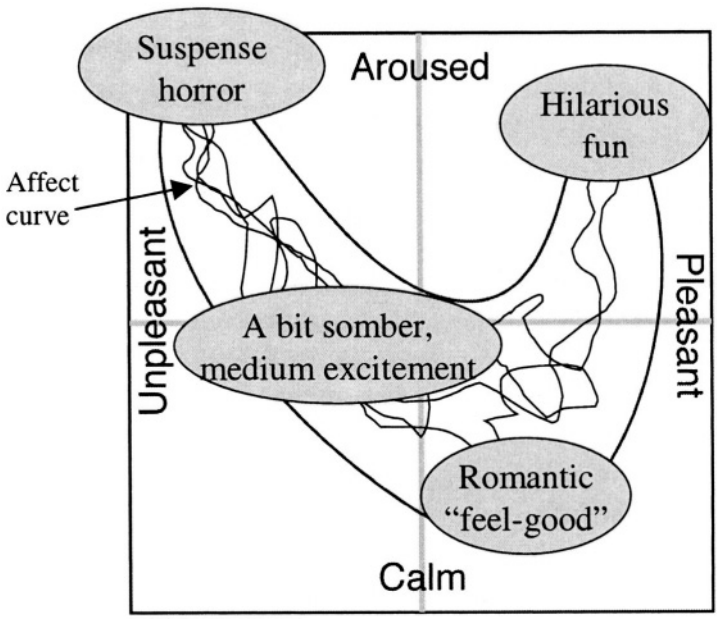

Figure 5-4. The content of a video segment can be indexed automatically by the label that characterizes the area of the 2D affect space through which the part of the affect curve corresponding to that segment passes.

5.4 AFFECTIVE VIDEO CONTENT MODELING

In order to obtain the affective content representation as described in the previous section, models need to be developed for the arousal and valence time curve. These models fulfill the task of deriving arousal and valence values from the low-level features computed in a video.

5.4.1 Criteria for model development

As arousal and valence are psychological categories, their models need to be psychologically justifiable. To achieve this, we can introduce the following three criteria that a model for the arousal, valence or affect curve should satisfy [Han04]:

- Comparability
- Compatibility
- Smoothness

The first criterion *(Comparability)* ensures that the values of the arousal, valence and the resulting affect curve obtained in different videos for similar types of events are comparable. This criterion obviously imposes normalization and scaling requirements when computing the time curves. The second criterion *(Compatibility)* ensures that the affect curve traverses an area in the valence-arousal coordinate system, the shape of which roughly corresponds to the parabolic-like contour of the 2D affect space. The third criterion *(Smoothness)* accounts for the degree of memory retention of preceding frames and shots [Ada00]. It ensures that the perception of the content, and consequently the mediated affective state, does not change abruptly from one video frame to another but is a function of a number of consecutive frames (shots).

5.4.2 How to select features?

Little is known regarding the relations between the low-level features and affect. While the problem of bridging the semantic gap remains very hard in the case of cognitive video content analysis, the magnitude of this problem in the affective case is even larger. The reason for this is that in the cognitive case the low-level features describe aspects of a real entity, like, for instance, the choice of the color *red* as one of the features to characterize a red car. In the affective case, however, we need to relate the low-level features to something rather abstract, such as feeling or emotion: which color combination, sound or event is to be related to happiness, disgust or fear?

5.4.2.1 Visual features

Colombo et al. [Col99] elaborate on the effects of color combinations in art images on the human affective state. Red color is found to communicate happiness, dynamism and power. Orange is thought to resemble glory, green should elicit calmness and relaxation, while blue may suggest gentleness,

fairness, faithfulness and virtue. Purple, on the other hand, sometimes communicates fear, while brown is often used as the background color for generating relaxing scenes. Further, a sense of uneasiness can be evoked by the absence of contrasting hues and the presence of a single dominant color region. This effect may also be amplified by the presence of dark yellow and purple colors. As opposed to this, the sense of calmness and quietness can be conveyed by combining complementary colors.

As reported in [Col99], the effect of color has been used in advertising practice to induce product-related mood effects in the potential customers [Haa88]. Very often, color is also combined with the elements of scene structure to enhance the effectiveness of advertising. This structure is mainly determined by the main edges found in a video frame, which are defined by the camera angle and the properties of the objects in the scene. For instance, the dominance of oblique lines in the scene communicates dynamism and action, while flat lines (e.g. the horizon) induce a sense of calmness and relaxation. Also, saturated colors can be used in combination with specific camera angles to communicate a sensation of dynamism.

One of the most extensively investigated visual features in the context of affective video content analysis is motion. Research results show that motion in a television picture has a significant impact on individual affective responses. This has been realized also by film theorists who contend that motion is highly expressive and is able "to evoke strong emotional responses in viewers" [Arn83, Gia76]. In particular, Detenber et al. [Det97] and Simmons et al. [Sim99] investigated the influence of camera and object motion on emotional responses of humans and concluded that an increase of motion intensity on the screen causes an increase in arousal, and thus also in the magnitude of valence. The sign of valence is, however, independent of motion: if the mood of a test person was "positive" or "negative" while watching a still picture, the type of the mood will not change if a motion is introduced within that picture.

5.4.2.2 Vocal features

In an analysis of the studies performed by Davitz [Dav64], Pittam, Gallois and Callanite [Pit90], and Chung [Chu95], Rosalind Picard [Pic97a] discusses a variety of vocal features that have been proposed so far in the attempts to enable computers to recognize affect from speech. The changes in arousal seem to be correlated, for instance, with pitch range, loudness, spectral energy in higher frequencies (up to 4kHz), and speech rate (e.g. faster for fear or joy and slower for disgust or romance). The sign of valence, however, is believed to be communicated by more subtle and more complex speech properties, such as inflection, rhythm, duration of the last syllable of

a sentence (short in anger and long in joy and tenderness), and voice quality (more resonant for joy and tenderness, and breathy for anger and sorrow). Further, the studies of Williams and Stevens [Wil69, Wil72] were mentioned, where the features, such as fundamental frequency contour, average speech spectrum and precision of articulation, were used to discriminate the affective states of fear, anger and sorrow. Finally, linear predictive coding parameters of speech were used together with speech power and pitch to recognize eight affective states (fear, anger, sadness, joy, disgust, surprise, teasing, neutral) in persons interacting with an animated character [Tos96].

Table 5-1. Summary of human vocal emotion effects. The effects described are those most commonly associated with the emotions indicated, and are relative to neutral speech (adopted from Murray and Arnott [Mur93])

	Anger	*Happiness*	*Sadness*	*Fear*	*Disgust*
Speech rate	Slightly faster	Faster or slower	Slightly slower	Much faster	Very much slower
Pitch average	Very much higher	Much higher	Slightly lower	Very much higher	Very much lower
Pitch range	Much wider	Much wider	Slightly narrower	Much wider	Slightly wider
Intensity	Higher	Higher	Lower	Normal	Lower
Voice quality	Breathy chest tone	Breathy blaring	Resonant	Irregular voicing	Grumbled chest tone
Pitch changes	Abrupt on stressed syllables	Smooth upward inflections	Downward inflections	Normal	Wide downward terminal inflections
Articulation	Tense	Normal	Slurring	Precise	Normal

An idea about the possibilities to relate the properties of speech to human affective states can also be obtained from the study of Murray and Arnott [Mur93] on human vocal emotion. They analyzed a wide range of comments made in literature about vocal effects caused by particular emotions. A high consistency of these comments provided the base for summarizing the vocal effects as shown in Table 5-1 for five primary emotions: anger, happiness, sadness, fear and disgust. For further information the reader may want to refer to the study article [Mur93].

In spite of the numerous studies on the relations between vocal effects and affect, inferring the affect from speech in practice is still an unresolved problem. The difficulty of this problem can best be seen in the fact that, when tested on neutral speech or on speech with obscured meaning, affect is properly recognized by humans only in 60% of all cases [Sch81]. Clearly, the question may arise whether we ask the computer to perform an impossible task. The results of affect recognition by humans show, however, that a person can usually distinguish arousal in the voice (e.g. angry versus sad), while the main obstacle is to properly identify the type (valence) of the affective state of the speaker. One may, therefore, conclude that the ambiguity in speech properties regarding the arousal recognition is relatively low, and that this detection task could be automated with reasonable success. In order to be able to do the same for valence, however, we need to also consider the context (scene, event), the content of the speech, and the features of other modalities (e.g. visual).

5.4.2.3 Editing-related features

By adjusting the length of the shots in relation to one another, a filmmaker is able to control the rhythmic potential of editing [Bor01]. The patterning of shot lengths [Ada00] is therefore a popular tool for the director to create the desired pace of action (e.g. in a movie). The director typically chooses for shorter shot lengths in movie segments that are to be perceived by the viewers as those with a high tempo of action development, or to create stressed, accented moments. As opposed to this, longer shots are typically used to de-accentuate an action. We illustrate this on the example from [Bor01]: "In editing Raiders of the Lost Ark, Steven Spielberg discovered that after Indiana Jones shoots the gigantic swordsman, several seconds had to be added to allow the audience's reaction to die down before the action could resume". In this sense, the varying shot lengths can be linked to the intended changes in the magnitude of arousal that is evoked in the audience along a movie. Note that regarding the pace at which the video content is offered to a viewer, an increase in shot-change rate is likely to have a similar impact on a viewer's arousal as an increase in the overall motion activity.

Wide variations in shots lengths can also be a good indication of how the director of a live broadcast responds to interesting events. We can explain this on the example of a soccer match that is broadcasted most of the time using one camera that covers the entire field. The director switches from one to another camera (e.g. by zooming onto a particular event, the bench or the spectators) only occasionally, which results in rather long shots. However, whenever there is a goal, or an important break (e.g. due to foul play, free

kick, etc.), the director immediately increases the rate of shot changes trying to show everything that is happening on the field and among the spectators at that moment. In this way, any increase in shot-change rate during a live broadcast is likely to be related to the director's response to an increase in the general arousal evoked in the sport arena.

5.4.3 An example approach to modeling arousal time curve

To illustrate the possibilities for approaching the development of affect models introduced above, we will now briefly discuss the method for arousal time curve modeling that was originally proposed by Hanjalic and Xu in [Han01] and slightly modified in [Han04]. Although being rather simple, this method already provides a reasonable correlation between the curve behavior and the arousal-related aspect of the video content, or, in other words, the expected amount of excitement that is evoked in the user while watching the video.

5.4.3.1 A general arousal model

We approach the arousal modeling by considering the function $G_i(k)$ that models the arousal over the frames k as revealed by the feature i. This function can be interpreted as one of the elementary components (primitives) of the arousal time curve. Namely, it is unrealistic to expect that a single feature can reveal the complete variations of arousal along a video. For instance, an increase in arousal during a soccer television broadcast is detectable at some places through the cheering crowd (changes in sound energy) and at some other places through an increase in shot-change rate (e.g. a break due to a foul play). Therefore, we model the arousal time curve $A(k)$ in general as a function of N components $G_i(k)$:

$$A(k) = F(G_i(k), i = 1,.., N) \tag{5.1}$$

Here, the function F serves to integrate the contributions of all the components $G_i(k)$ in the overall course of arousal along a video. In order for the function F to satisfy the criteria of comparability and smoothness, these criteria need to be satisfied first by each component time function $G_i(k)$. This requirement can also be justified by the fact that each function $G_i(k)$ is an (elementary) arousal function by itself.

In order to obtain the component time curves $G_i(k)$ three arousal–related low-level features were selected:

- *The motion component,* obtained on the basis of the overall motion activity measured between consecutive video frames,

- *The rhythm component,* obtained by investigating the changes in shot lengths along the video,

- *The sound energy component,* obtained in synchronization with the video frame interval by computing the total energy in the sound track of a video.

The above features were selected to represent the arousal stimuli contained in different modalities of video (visual and audio) and those revealing the influence of video authoring (editing). In this sense, we expect the contributions to the course of arousal originating from these features to be largely independent of each other.

5.4.3.2 The motion component

We start the computation of the motion component of the arousal function (5.1) by computing the motion activity $m(k)$ at each video frame k. Motion vectors are computed using the standard block-based motion estimation between two adjacent frames k and $k+1$. The motion activity value is then found as the average magnitude of all (B in total) motion vectors $\boldsymbol{v}_i(k)$, normalized by the maximum possible length of a motion vector $|\boldsymbol{v}_{\max}|$.

$$m(k) = \frac{100}{B\,|\boldsymbol{v}_{\max}|}\left(\sum_{i=1}^{B}|\boldsymbol{v}_i(k)|\right)\% \qquad (5.2)$$

Note that the motion activity values (5.2) are scaled to the range between 0 and 100 % – a range that will be imposed also for other model components so they can be combined with each other on the same basis, but also for the resulting arousal levels to be expressed in percentages. In this way, we create a solid basis for the fulfillment of the *Comparability* criterion.

In view of the *Smoothness* criterion, the obtained motion activity time curve is not directly suitable for being a component of the arousal model. First, the value (5.2) may quickly fluctuate within the same shot. Second, motion activity values may fluctuate in different ranges for two consecutive shots (e.g. total motion activity within a close-up shot is much larger than that in a shot taken from a large distance) which results in "jumps" of these values from one range to another at shot boundaries. Third, measuring motion activity for the consecutive frames will encounter unavoidably the

high peaks or other noises at shot boundaries and locations of other editing effects as well. In order to fulfill the *Smoothness* criterion the *m(k)* is convolved with a sufficiently long smoothing window. Hanjalic and Xu use the Kaiser window $K(l_1, \beta_1)$ of the length l_1 and the shape parameter β_1 for this purpose.

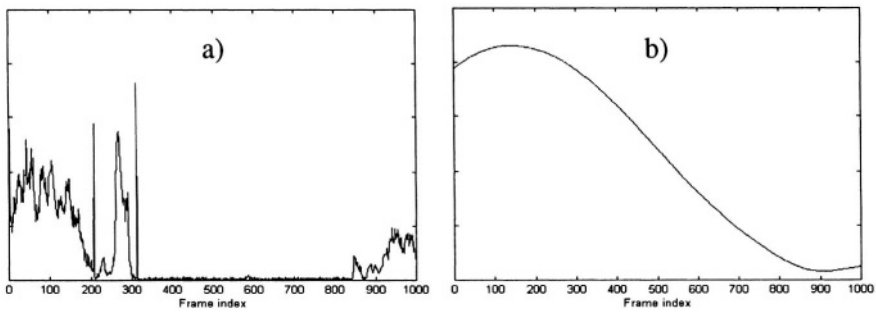

Figure 5-5. (a) The raw motion activity and (b) after smoothing has been applied

We demonstrate the effect of the smoothing operation in Figure 5-5, where a video segment consisting of three consecutive shots of a typical soccer match is considered. The two shot boundaries can be easily recognized as the sharp peaks around frames 200 and 300 of the motion activity function *m(k)* in Figure 5-5a. The first and the second shot are characterized by a high motion activity, corresponding to close-up shots of players running on the field. The third shot was taken by a camera mounted on a high ground with a wide view of the field, hence the overall motion activity obtained is rather low initially. The changes toward the end of the third shot take place when the camera is maneuvered to view the previously covered part of the field in the course of the game. The first two shots belong to an exciting segment of a soccer broadcast (goal chance). Starting from the second shot change, the game becomes stable and the level of excitement decreases. However, the increase and decrease of a user's excitement cannot change abruptly. While a user's excitement will reach its peak somewhere during the series of close-up shots, it will start to descend gradually, as the game becomes stable. Gradual reduction in the level of excitement will continue after the second shot change since the user needs time to recover from previous exciting events. Finally, when the course of the game becomes more dynamical around frame 850, the excitement of the user will start to rise, though with a certain delay - again due to the inertia of human affective states. As can be seen from Figure 5-5b, the smoothed motion activity curve is much more likely to mimic the variations in a user's excitement, as described above.

We adopt the smoothed motion activity curve as the motion component $G_1(k)$ of the arousal time curve (5.1). We represent this component analytically as

$$G_1(k) = \frac{\max_k(m(k))}{\max_k(\tilde{m}(k))}\tilde{m}(k)\ \% \tag{5.3}$$

Here, $\tilde{m}(k)$ is the result of the convolution of the curve *m(k)* with a smoothing window, that is $\tilde{m}(k) = m(k) * K(l_1, \beta_1)$. Scaling the curve $\tilde{m}(k)$, as indicated in (5.3), serves to put the values $G_1(k)$ back inside the original value range (0-100%).

5.4.3.3 The rhythm component

Similar to the analysis of the motion activity, in the following we aim at obtaining a curve that is a function of the frame index *k* and that reveals a connection between a viewer's arousal and the time-varying shot lengths. We start modeling the influence of the shot-change rate on a viewer's arousal by defining the function *c(k):*

$$c(k) = 100\, e^{\left(\frac{1-(n(k)-p(k))}{\delta}\right)}\ \% \tag{5.4}$$

Here, *p(k)* and *n(k)* are the positions (frame indexes) of the two closest shot boundaries to the left (beginning of the shot) and right (end of the shot) of the frame *k,* respectively, and the parameter δ is the constant determining the way the *c(k)* values are distributed on the scale between 0 and 100 %. As illustrated by the example in Figure 5-6a, the curve *c(k)* is typically a step curve, with each step corresponding to video segment between two shot boundaries and with the height of each step being inversely related to the interval between the boundaries: the shorter the interval the higher the value *c(k).* Again, due to incompatibility of vertical edges in *c(k)* with the *Smoothness* criterion, we convolve the *c(k)* curve with the same smoothing window as in the case of motion activity. Scaling the convolution result back to the original value range results in the function that we adopt as the rhythm component $G_2(k)$ of our arousal model (5.1), and that is illustrated by the example in Figure 5-6b:

$$G_2(k) = \frac{\max_k(c(k))}{\max_k(\tilde{c}(k))}\tilde{c}(k)\ \% \quad \text{where} \quad \tilde{c}(k) = K(l_1, \beta_1) * c(k) \tag{5.5}$$

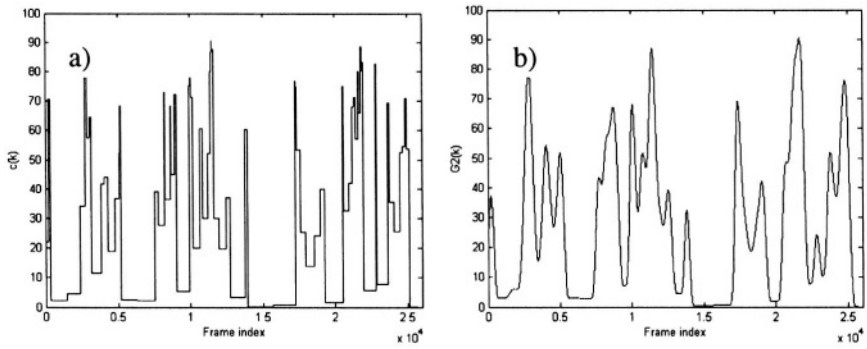

Figure 5-6. (a) An example of the curve *c(k),* (b) The corresponding curve $G_2(k)$

5.4.3.4 The sound energy component

As the third component of the arousal model (5.1) the sound energy contained in the audio track of a program is considered. One energy value is computed for the time length of each video frame. Thus the number *s* of audio samples used to compute this value is determined as the ratio between the audio sampling frequency (typically 44.1 kHz for CD quality) and the video frame rate. The power spectrum is computed for each consecutive segment of the audio signal containing *s* samples. An equivalent of the sound energy value *e(k)* is then computed by adding up all spectral values.

We again apply the same Kaiser window as in previous sections to smooth out the originally "rough" time curve *e*(*k*). However, unlike the other two arousal components, sound energy is dependent on the volume level at which the audio track is recorded. Since neglecting this fact would result in sound energy time curves that are not comparable over different videos, we proceed as follows. First, we scale the energy time curve obtained after convolution to the range between 0 and 1. Then, we weight the obtained curve according to its mean value. If the curve is characterized by only a few highly distinguishable peaks, then its mean value is lower than in the case where the curve homogeneously covers the entire value range. Since in the first case it is likely that the video contains several highly exciting events, these peaks should play a significant role in shaping the final arousal time curve. In the second case, however, the presence of exciting events is uncertain. Then, due to ambiguity related to the recording volume level, the influence of the energy component on shaping the arousal time curve is kept limited. With this in mind, with $\tilde{e}(k) = K(l, \beta) * e(k)$ and with *W* being the length of the analyzed video in frames, we define the sound energy component $G_3(k)$ of our arousal model as follows:

$$G_3(k) = 100\, e_n(k)\,(1-\bar{e}_n)\ \%$$

where (5.6)

$$e_n(k) = \frac{\tilde{e}(k)}{\max_k(\tilde{e}(k))} \quad \text{and} \quad \bar{e}_n = \frac{1}{W}\sum_k e_n(k)$$

5.4.3.5 Arousal as a weighted average of the components

Figure 5-7 shows all three arousal components computed for an excerpt from a soccer match. When compared with the content description of characteristic segments of this excerpt (see labels), one can see that at the times of exciting events (goals, goal chances, breaks), a distinguished local maximum can be found in at least one of the component time curves, as opposed to less exciting segments. One can also notice that these local maxima are not necessarily aligned. For instance, in the case of a score, the following scenario is possible: the spectators first cheer the action (sound energy peak), then there are cameras zooming in on running players (motion activity peak) and, finally, there are cameras zooming in on the teams' benches and the spectators (cut density peak). This fact motivates the definition of the function F as a weighted average of the three components, which is then convolved with a sufficiently long smoothing window in order to merge neighboring local maxima of the components into one peak and so to compensate for the possible asynchrony of the behavior of the component time curves at places of exciting events [Han01, Han04]. The result is finally re-scaled to the 0-100% range:

$$A(k) = \frac{\max_k(a(k))}{\max_k(\tilde{a}(k))}\,\tilde{a}(k)\ \%$$

with (5.7)

$$a(k) = \sum_i \eta_i G_i(k), \quad \sum_i \eta_i = 1 \quad \text{and} \quad \tilde{a}(k) = K(l_2, \beta_2) * a(k)$$

Here, η_i are the coefficients weighting the component functions $G_i(k)$. The convolution is performed again with the Kaiser window. However, as indicated by the values l_2 and β_2 this window may have a different length and shape parameter compared to the window used previously for the three components.

5.4.3.6 Model performance and utilization

Having described the example method for modeling the arousal components $G_i(k)$, and for integrating these components as in (5.7) to form a complete arousal model, we now proceed to illustrate the performance of the obtained arousal model on real media data. The choice of test video sequences was based on two considerations. First, in order to obtain meaningful results, the sequences selected should be those in which the changes in arousal are most likely induced by the stimuli depicted by the low-level features adopted. Second, the sequences selected should be characterized by the content flow on which an average user is expected to react in a "standard way" in terms of arousal. For instance, the arousal is expected to rise when the development of a soccer game goes from the stationary ball exchange in the middle of the field and finishes via a surprise fast action with the goal. In the same fashion, the arousal is supposed to decrease with the stabilization of a situation in an action movie, following a rapid action event.

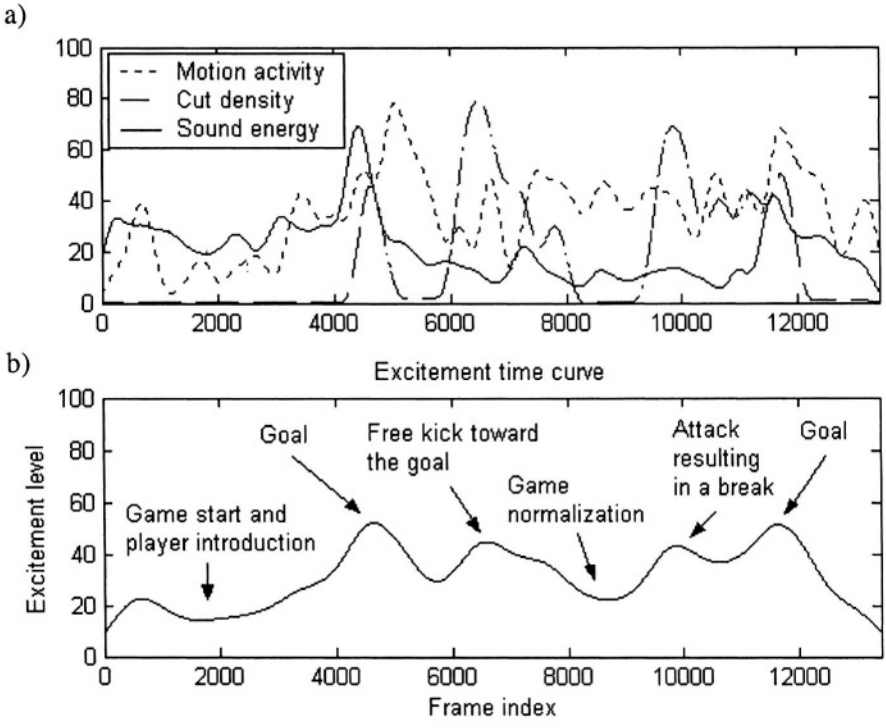

Figure 5-7. (a) The component time curves and (b) the resulting arousal time curve obtained for an excerpt from a soccer match

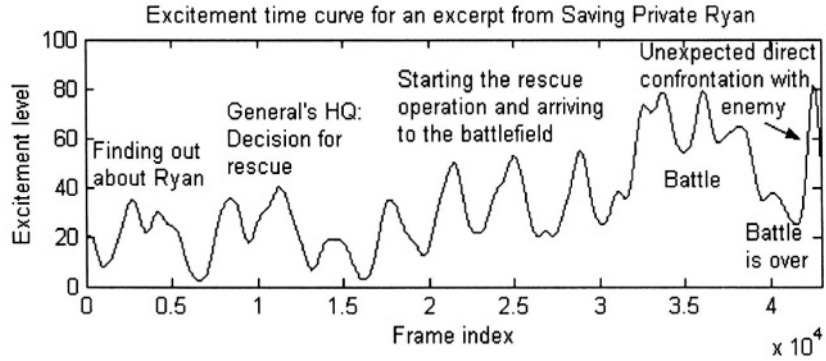

Figure 5-8. Arousal time curve obtained for an excerpt from a movie

Figures 5-7 to 5-10 illustrate the model performance on the test sequence set including excerpts from two different soccer matches (and two different broadcasters) and from the movies "Saving Private Ryan" and "Jurassic Park 3". For each test sequence, the same set of parameter values was used: the pixel block size for motion estimation was selected as 16, the coefficients η_i were selected as 1/3, and δ was set to 300. The length and shape parameter of the Kaiser window used for arousal components were 700 and 5, and those for the complete arousal model were 1500 and 5, respectively. In each figure the characteristic segments are labeled to reveal the actual content of the corresponding video such that the model performance can be judged. In each figure the reader may observe the behavior of the arousal time curve in the global sense, and whether it complies with the content development along various sequence segments. The reader may also check the similarity of the arousal levels obtained for similar events in different sequences.

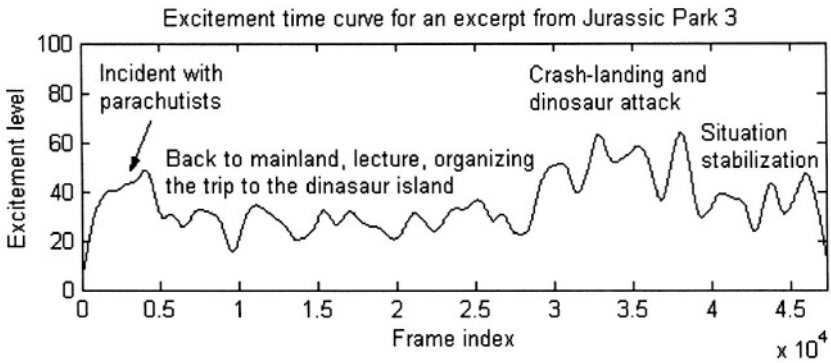

Figure 5-9. Arousal time curve obtained for an excerpt from a movie

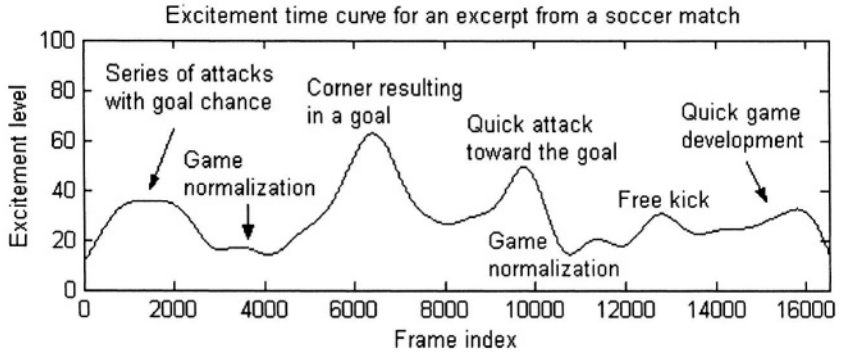

Figure 5-10. Arousal time curves obtained for an excerpt from a soccer match

Clearly, a high correlation can be observed between the behavior of the arousal time curve in each figure and the content development of a test sequence used. We emphasize, however, that although good results could be obtained for two different video genres (soccer and movie), it is unrealistic to assume the constancy of the arousal feature set across a broad scope of video genres. This is simply due to the fact that the features revealing the arousal stimuli in one genre may not be present in another genre or may not be discriminative enough for arousal measurement in that genre.

5.4.4 An example approach to modeling valence time curve

The *Compatibility* criterion described in Section 5.4.1 requires that the affect curve generated through combining the arousal and valence time curves should cover an area in the valence-arousal coordinate system that has a parabolic-like shape resembling the 2D affect space (Figure 5-2). Clearly, this criterion confines that the values of arousal and the absolute values of valence are related to each other, which means that in general the range of arousal values determines the range of absolute valence values. We could therefore start the development of a valence model by defining the function *r(k)* that captures this value range dependence [Han04].

$$r(k) = a(k) \cdot sign\{H(D_j(k), j = 1,..,M)\} \tag{5.8}$$

Here, *k* is again the frame index, and *a(k),* as defined in (5.7), is the arousal function before smoothing. Similar to the discussion on the arousal model (5.1), each component $D_j(k)$ in (5.8) models the changes in valence as

revealed by the feature *j,* while the function *H* serves to integrate the contributions of all the components in the final valence time curve. Clearly, the values *r(k)* are determined solely by the values of the arousal, while the function *H* only determines the sign of *r(k).*

The values of *H* are used again in the next step to compute the variations of the valence in the value range specified by the arousal. In order for the valence values to remain in the proper range, the amplitude of these variations needs to be much smaller than the value of the arousal determining that range. With this in mind we define the variance function *g(k)* as follows:

$$g(k) = \frac{n}{100} \cdot \max_k A(k) \cdot \frac{H(D_j(k), j = 1,.., M)}{\max_k \left| H(D_j(k), j = 1,.., M) \right|} \quad \% \tag{5.9}$$

The number *n* determines the magnitude of allowable variations of valence values in the range specified by the arousal. As shown in (5.9), this magnitude is not allowed to exceed *n* percent of the maximum arousal value.

We now model the valence time curve as

$$V(k) = \frac{\max_k \left| v(k) \right|}{\max_k \left| \tilde{v}(k) \right|} \tilde{v}(k) \, \%$$

with (5.10)

$$v(k) = r(k) + g(k) \quad \text{and} \quad \tilde{v}(k) = K(l_2, \beta_2) * v(k)$$

The smoothing window used here is the same as the one used for smoothing the final arousal time curve in (5.7). The main purpose of smoothing the curve *v(k)* is to eliminate jumps appeared in the *r(k)* curve due to the sign change in (5.8).

As is clear from discussions above, the role of function *H* is actually analogous to that of function *F* in (5.1). Therefore, the search for the proper form of function *H* can be done in the similar way as for function *F.* In the following, we first describe how to model a component function $D_j(k)$ using one of the valence-related features – the *pitch average* – such that it satisfies the criteria of comparability and smoothness. We then demonstrate the concept of modeling the valence time curve as explained above based on the example of the simple curve derived from the pitch-average component.

5.4.4.1 The pitch-average component

We compute the pitch signal using the off-the-shelf software and average the pitch values temporally over each video segment of length *L*. This results in the pitch-average time curve *P(k)*. As studied by Murray and Arnott [Mur93], the average pitch can be useful in distinguishing between some positive and negative affective states, such as "happiness" (high pitch average) and "sadness" (low pitch average).

In order to associate the average pitch value with a corresponding valence value that may also be negative, we define the following function:

$$p(k) = P(k) - N \tag{5.11}$$

Here *N* is what we call the "neutral feeling" frequency, and serves to map the low (high) values of the pitch average to the corresponding negative (positive) valence values.

In view of the smoothness criterion, the function (5.11) is not directly suitable to serve as a valence component time curve due to its step-wise nature. We therefore smooth the values (5.11) using the same Kaiser window as in the case of the arousal components. The result is the pitch-average component $D_1(k)$ of the valence time curve:

$$D_1(k) = \frac{\max_k |p(k)|}{\max_k |\tilde{p}(k)|} \tilde{p}(k) \ \% \quad \text{with} \quad \tilde{p}(k) = K(l_1, \beta_1) * p(k) \tag{5.12}$$

5.4.4.2 Model performance

The measurement of the affect type (valence) is much more ambiguous than the measurement of the affect intensity (arousal). We therefore choose to evaluate the valence model (5.10) in its simplest form, where the function *H* is based on one component function only, that is, $H(D_j(k), j=1,\ldots,M) = D_1(k)$, and in a controlled situation. The purpose of evaluation in this section is to prove the concept of

- modeling the valence components as shown by the example (5.12),
- modeling the valence time curve along the steps (5.8-5.10),
- generating the affect curve on the basis of the corresponding arousal and valence curves, as explained in Section 5.3.2.

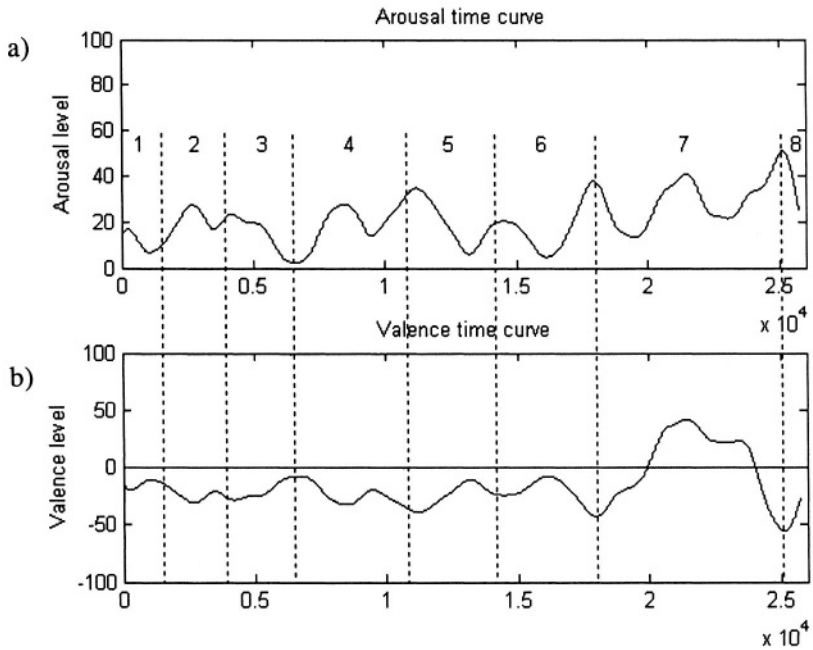

Figure 5-11. (a) The arousal curve obtained for an excerpt from the movie "Saving Private Ryan", (b) The valence curve obtained for the same excerpt on the basis of the pitch-average component (5.12)

Table 5-2. Labels describing the content of the test sequence in Figure 5-11

Segment	Content description
1	US Army HQ, typists make letters for soldiers' families, male voices reading the letters
2	Colonel's office, finding out about private Ryan
3	Bad news brought to Ryan's home
4	General's office, decision is being made to search for Ryan
5	Omaha Beach, US Army HQ, an officer gets the order to search for Ryan
6	Omaha Beach, US Army HQ, preparation for the search action
7	Beginning of the action, walking through the fields
8	It starts to rain and gets dark, the suspense grows, the actual beginning of the action

Since we chose $D_1(k)$ as the pitch-average component, we select a test video sequence such that its affective content can largely be determined on the basis of the pitch average only. For this purpose we selected an excerpt from the movie "Saving Private Ryan" where the sound track consists of male voices that are only sporadically interrupted by noise or music.

Figure 5.11 shows the arousal and valence time curve obtained for the selected test sequence. Besides the parameters already specified in Section 5.4.3, additional parameters here are the "neutral feeling" frequency N that is set to 150 Hz [Pic97b], the value of n that is set to 10, and the pitch-average segment length L that is set to 909 frames. The rather odd value that we used for L resulted from our attempt to partition the test sequence into the segments of equal length, which are, at the same time, synchronized with the segments selected by the off-the-shelf software to compute the pitch. In order to evaluate the correlation between the obtained arousal and valence values and the actual content of the sequence, we have labeled different parts of the sequence to describe their contents in as much detail as possible. These labels can be found in Table 5-2.

Figure 5-11a shows that the changes in arousal are not that strong. This was expected as the entire sequence is rather stationary and mainly contains conversations. A slight increase of the average arousal value in the segment 2, 4 and 7 as compared to the previous segment, is, however, quite correlated to the actual content development of the sequence along these segments.

The range of the valence values in Figure 5-11b indicates that the valence time curve is basically a scaled (and mirrored, where negative) version of the arousal curve, on which the allowed variations modeled using the pitch-average component are superimposed. The first interesting spot in Figure 5-11 is the switch of the valence curve from the negative to positive values around the frame 20000. This switch reveals the change in the prevailing mood from mostly somber in the first part of the sequence to a "casual" every-day mood and even some happiness. This is then followed by, again, expected switch of the curve to the range of negative valence in the segment 8. The course of the obtained valence time curve largely corresponds to expectations. However, the simplicity of the function H has also lead to slight imperfections in the obtained curve. Namely, the segments 5 and 6 also contain parts that are characterized by the similar "casual" every-day mood as in the segment 7. These parts are not properly revealed by the valence time curve in Figure 5-11b.

We now combine the arousal and valence curve from Figure 5-11 in the affect curve that provides the complete representation of the affective content of the video clip under study. The parabolic shape of this curve shown in Figure 5-12 clearly indicates the compatibility of the obtained curve with the 2D affect space. As we can read from the curve, the

prevailing mood of the test sequence is rather somber (low-to-medium arousal and negative valence) with the exception of one segment that is characterized by a mid-level arousal and a positive valence.

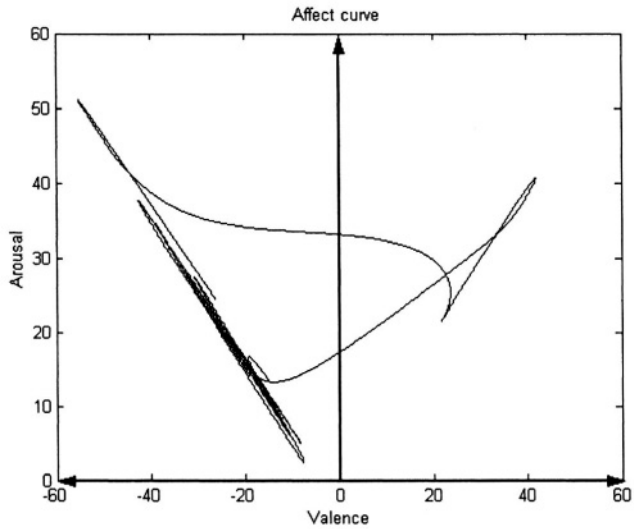

Figure 5-12. The affect curve obtained by combining the curves from Figure 11a-b for n= 10

5.5 APPLICATIONS

As we mentioned in the introduction to this chapter, the ability to extract affective content from a video will not only extend the scope of the possibilities for video indexing and retrieval, but is also likely to provide valuable information that can be used in the process of personalizing the video delivery to the user. In this section we will discuss these new possibilities in more detail.

5.5.1 Automatic video indexing using affective labels

By computing the arousal (A) and valence (V) values from the features of a given temporal video segment the obtained (A,V) pairs can be mapped onto a particular area in the 2D affect space. Then, the label characterizing the affective states in this area can be used to describe the affective content of this video segment, as illustrated in Figure 5-13. After the labels have been assigned along a video, it is easy to retrieve the segments characterized by a particular "mood", but also to leave them out if found unsuitable for the audience (censorship). An efficient way of indexing the entire video

following the procedure in Figure 5-13 is to compute the affect curve for the entire video first and then to let the video segments be indexed automatically as we explained in Section 5.3.2 (Figure 5-4).

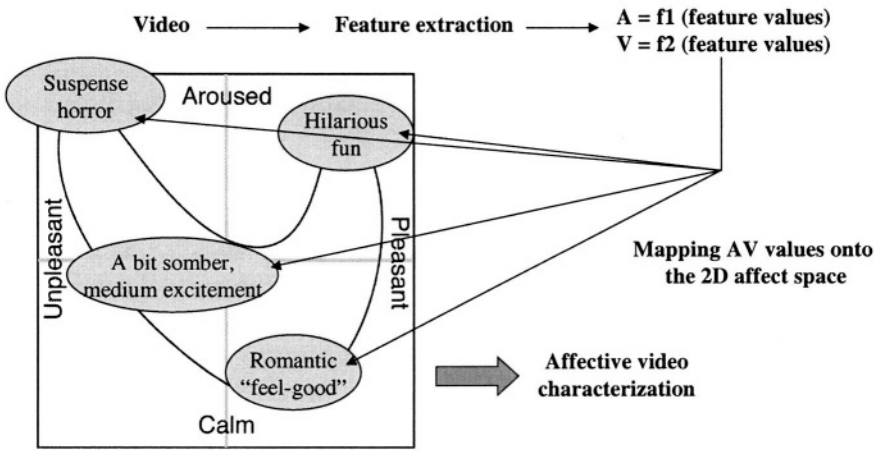

Figure 5-13. Quantifying the affective aspects of the video content

5.5.2 Highlights extraction

Although the highlights generally stand for the most interesting parts of a video, the definition of what is "interesting" may strongly vary across diverse video genres and for different applications. For instance, while a highlight of a news program is determined by the novelty and impact of the news (e.g. "breaking news", "headline news"), the criteria for highlight extraction from a home video are rather content-dependent, like "where my baby walked for the first time". The ability to analyze video at affective level will broaden the possibilities for highlights extraction in a number of new application contexts, such as automated movie trailer generation and sport broadcast pruning.

5.5.2.1 Automated movie trailer generation

Movie producers hope to attract large audience to cinemas or to video on-demand services by advertising the movies using trailers. A trailer is a concatenation of movie excerpts that last only for several minutes but are capable of commanding the attention of a large number of potential cinema goers and video on-demand users. Analyzing a movie at affective level can

provide valuable clues about which parts of the movie are most suitable for being an element of the trailer. This is because emotion plays a primary role when processing mediated stimuli [Die99]. The emotion (affective content) influences the attention of a user and his evaluation and memory for the mediated facts (cognitive content). Consequently, the perception of the affective content interferes with the perception of the cognitive content and influences user's reactions to the cognitive content, such as liking or not-liking, enjoyment and memory. Further, since memory is one of the most important factors when creating a trailer, it is worthy to notice that memory for highly emotional video fragments has been proven to last longer than the memory for non- or less-emotional video clips [Lan95b, Lan96]. Therefore, having available the algorithms for video analysis at the affective level, the creation of movie trailers could be performed fully automatically.

5.5.2.2 Automated pruning of a sport TV broadcast

The idea of automatically creating trailers for movies can easily be extended to the case of sport programs as well. The sport events advertise themselves among the TV viewers using the "most touching scenes in the sport arena" with the objective of selling their commercial blocks as profitably as possible. However, the availability of the algorithms for automatically extracting sport program highlights becomes handy also in the process of *pruning* a large volume of recorded sport video: only the segments being worth watching are kept, while the remaining, less interesting parts are discarded. The sports programs are particularly interesting objects for pruning as they lack a story line, and as the events being worth watching (e.g. goals in soccer, home runs in baseball, touchdowns in football) are sparse and spread over a long period of time.

At this stage it is worthwhile emphasizing that the process of trailer generation may by much more subtle and complex than pruning. As the trailer serves to attract people to see a particular program, the video segments are searched for that are capable of influencing the affective state of the user correspondingly. In this sense, the process of trailer generation may involve an analysis of both the arousal and valence components of the affective video content. In the case of pruning, however, the user eventually has access to selected program segments only. In order not to discard any interesting segments, the filtering process should be less selective and preserve all segments of potential interest to the user.

The challenge of automatically pruning sport television broadcasts has been pursued widely in the scientific community in the past years [Li03]. An analysis of the previous work on this subject reveals that most of the approaches proposed so far are event-based: they are developed for a

particular sport genre and aim at detecting pre-defined events that are considered most interesting for that genre. Event detection is approached either by developing feature-based event models (e.g. [Gon95, Kaw98, Leo03, Sud98, Uts02, Xie02, Xu03]), by searching for keywords in speech (e.g. [Cha96]) and closed captions (e.g. [Nit00]), by using MPEG-7 metadata (e.g. [Jai02]) or by involving several of the abovementioned clues into inter-modal collaboration (e.g. [Bab03, Hua02, Sno03]). Clearly, the need for reliable event models makes the pruning process technically and semantically a complex task in many broadcasts. It also requires the development of a separate pruning algorithm for each particular sport program genre. Although event detection via keyword spotting may be performed in a more generic way for different sport program genres, the composition of the resulting video abstract is limited only to those events, for which obvious keywords are likely to be found. While this may be the case for the "goal" and "penalty" events in soccer or "home run" in baseball, other interesting events, such as a nice action on the net during a tennis game or a nice move of a goalkeeper in soccer, will probably remain undetected.

An alternative to the approaches discussed above is to search for a single event that is assumed to accompany an arbitrary highlighting event. For instance, Pan et al. [Pan01] based the detection of highlights on the detection of slow-motion segments. They observe namely that the interesting events – which ever these may be - are often replayed in slow motion immediately after they occur. Although being more generic than the methods discussed above, this highlights extraction mechanism may result in a large number of falsely extracted video segments: interesting events are often replayed in slow-motion during several later game breaks. Further, several slow-motion segments may be played after each other, showing the highlighting event from different camera angles. Similar problems emerge in the attempts to detect specific audio events, such as "applause", "cheering crowd" or "cheering commentator", that are often seen as indicators of the potential presence of a highlight. Although rather successful attempts to extract highlights based on an analysis of audio events alone were reported (e.g. [Xio03]), Rui et al. [Rui00] observe that such an approach, in general, is likely to lead to a large number of false alarms. The spectators and the commentator may, namely, become "loud" for the reasons that have nothing to do with the sport event considered (e.g. cheering a hot-air balloon that flies over the stadium). It is, therefore, not surprising that many proposed methods combine the "cheering" detection with other, in most cases event-specific clues (e.g. [Cha96, Dag01, Pet02, Rui00]).

A further step toward the development of generic tools for sport-program pruning is made by the approaches for detecting the high-level sport program structure. For instance, Li and Sezan [Li01] develop both a deterministic and probabilistic framework for detecting the "play" segments

of a sport video. These segments are assumed to contain all the interesting material of a program, as opposed to less interesting "non-play" segments. They demonstrate the applicability of their framework to football, baseball and sumo wrestling. An alternative approach to detecting "play" and "break" events was proposed by Ekin and Tekalp [Eki03], and was demonstrated on basketball, football, golf and soccer. However, although being very helpful in filtering out irrelevant video segments, these structure-oriented approaches are not suitable for highlights extraction: not all the material contained in a "play" segment can be considered a highlight. Further, the detection of structural elements as described above may not be possible in some sport disciplines, such as swimming or car racing.

Since it is realistic to assume that each highlighting event (e.g. goal, touchdown, home run, the finals of a swimming competition, or the last 50 meters in a running contest) induces a steady increase in user's excitement, we can search for highlights in those video segments that are expected to excite the user most [Han03], As the expected variations in a user's excitement induced by video can be modeled as a function of various non-content, and thus domain-independent video features, the affective content modeling provides a basis for developing a generic method for highlights extraction, independent of a sport program genre and the events contained there (Figure 5-14).

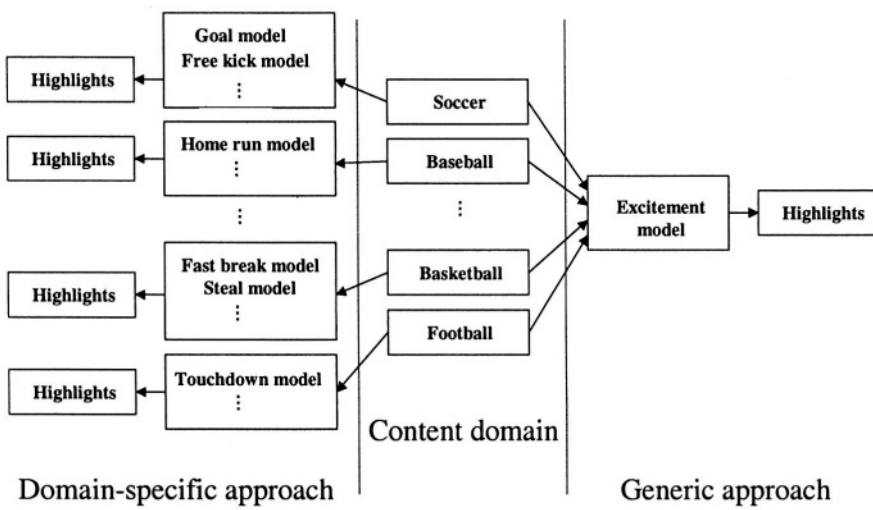

Figure 5-14. Generic versus domain-specific approach to sports highlights extraction

5.5.2.3 An example approach to sport program pruning

To illustrate the possibilities for the realization of the generic highlights extraction scheme in Figure 5-14, we recall the example approach to arousal modeling presented in Section 5.4.3. There we have shown how the time curves of three arousal components can be combined together into an arousal time curve. This curve mimics the influence of all three stimuli on user's arousal: there is a visible peak in the arousal time curve wherever at least one of the components reaches a significant local maximum. Then, given the arousal time curve *A(k)* and the maximum abstract length *L* in frames, the simplest approach to highlights extraction is to look at the values of the curve and to extract those video segments that are likely to excite the user most. To do this, we can draw a horizontal line cutting off the peaks of the curve in such a way that the number of frames in video segments where the peaks are found is not larger than *L*. This simple method for highlights extraction is illustrated in Figure 5-15. As the extraction process is driven by the local excitement level only, any event in a sport program may be included into the abstract, provided that the curve *A(k)* passes through a sufficiently high value range during that event. In this way, highlights are extracted in a generic fashion without the need for event modeling or artificially limiting the scope of the abstract content.

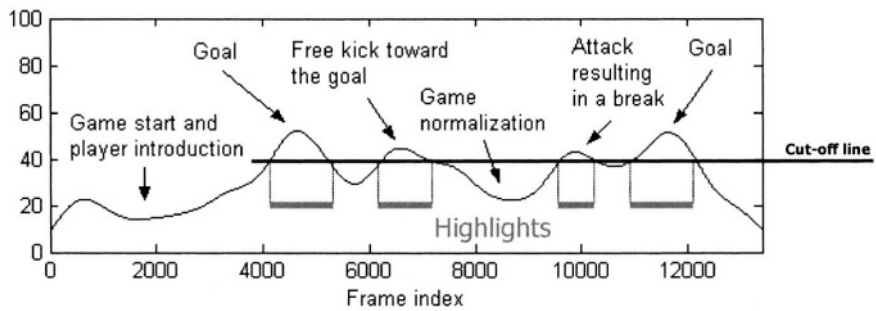

Figure 5-15. *Simple sports highlights extraction using the arousal time curve*

In view of the fact that each value of the curve *A(k)* results from multiple combined stimuli represented by the component time curves $G_i(k)$, we could also extract the highlights in a more sophisticated fashion, namely by taking into account the additional criterion of highlight "strength". The strength of a highlight can be defined as the number of component time curves $G_i(k)$ that rise to a high value range during this highlight. This number is therefore equivalent to the number of stimuli that have major influence on the

affective state of the user during a particular sport event, and can be said to determine the "richness" of the experience of that event: the richer the experience the stronger (better, more interesting) is the highlight. To ensure the selectiveness of the highlights extraction process, given the minimum allowed highlight strength *M,* only those video segments having the strength of at least *M* should be allowed to enter the process in the first place. This can be done by "filtering" the original time curve $A(k)$: the curve values in video segments that are likely to be the highlights of the required strength are left high while all other curve values are pulled down in order not to be captured by the cut-off line.

Let us consider the time stamp k and the values of the components $G_i(k)$ computed at that time stamp. We can now rank the values $G_i(k)$ in the descending order, denote the ranked values by $\overline{G}_j(k)$ and consider the elements of the subset $S_M = \{\overline{G}_j(k) \mid j = 1,..., M\}$. As the lower bound of the value range of all components in the subset S_M is determined by the value of the minimum of that subset, $\overline{G}_M(k)$, we can weight the values of the curves $G_i(k)$ in correspondence to this minimum, that is

$$G_i'(k) = G_i(k)\, w(k), \quad i = 1,.., N \tag{5.13a}$$

with

$$w(k) = f\left(\overline{G}_M(k)\right) \tag{5.13b}$$

The ideal behavior of the function $f(x)$ is illustrated by the curve in Figure 5-16. It secures gradual thresholding of the component time curves depending on the value of the argument $\overline{G}_M(k)$. As a possible analytical model for the function $f(x)$ the following expression based on the error function can be used, which is similar to the one we already introduced in Chapter 2:

$$f(\overline{G}_M(k)) = w(k) = \frac{1}{2}\left(1 + erf\left(\frac{\overline{G}_M(k) - d}{\sigma}\right)\right) \tag{5.14a}$$

with

$$erf(x) = \frac{2}{\pi}\int_0^x e^{-t^2} dt \tag{5.14b}$$

If the minimum of the subset S_M is not high enough, the weighting factor *w(k)* is close to zero. Consequently, the components $G_i(k)$ are pulled down, and therewith also the resulting excitement value *A(k)*. This is not the case only at those time stamps where the value $\overline{G}_M(k)$ is sufficiently high, meaning that the values of all components from the subset S_M are sufficiently high. There, the value of *w(k)* is close to one and the effect of filtering is negligible. Clearly, the filtering process is adaptive, as it is controlled by the parameter *M:* the higher the value of *M,* the more strict is the filtering of the arousal time curve.

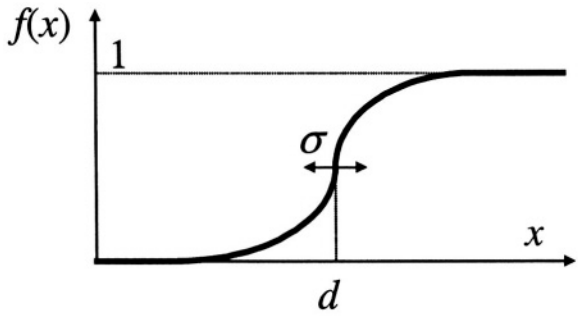

Figure 5-16. Ideal behavior of the function *w(k)*

By considering the processed components $G_i'(k)$ in the model (5.1), instead of the original components $G_i(k)$, we obtain a filtered version of the arousal time curve that we will refer to as *highlights time curve* $H_M(k)$:

$$H_M(k) = F(G_i(k), i = 1,..,N) \tag{5.15}$$

The time curve $H_M(k)$ could now serve, instead of the curve *A(k),* as the basis for extracting highlights using the methodology explained earlier in this section. By applying the cut-off line to the highlights time curve $H_M(k)$ only those video segments will be considered highlights, in which the excitement values remain high after the filtering process (5.13). Consequently, the parameter *M* controlling the filtering process can also be said to determine the composition of the highlighting video abstract.

Figure 5-17 illustrates the possibilities for practical implementation of the adaptive highlights extraction method outlined above. While the filtering process (5.13), the formation of the highlights time curve and the actual highlights extraction using a cut-off line can be considered the same for all sport program genres, this is not necessarily true for the feature set used to

compute the component time curves $G_i(k)$. This, however, is not a problem as the detection of program genre can be done using, for instance, the side information accompanying the broadcast (e.g. Electronic Program Guide (EPG)), or by performing the program genre classification locally, using a suitable algorithm that can be developed with the techniques we already discussed in Chapter 4.

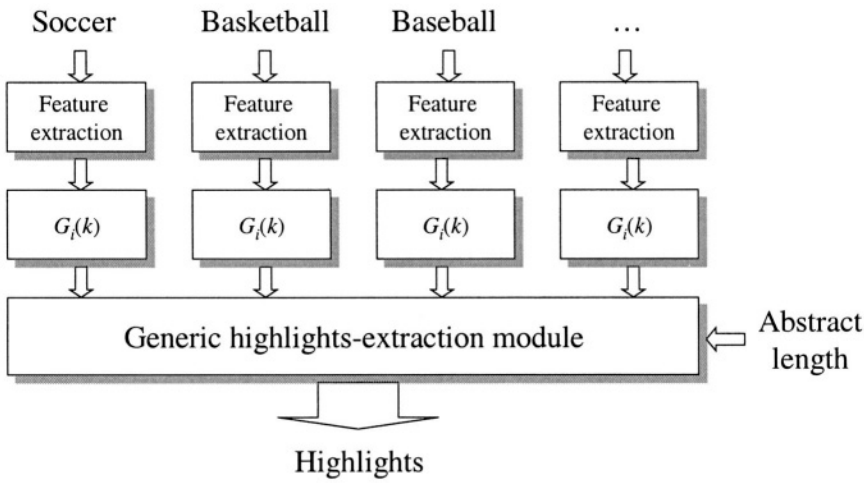

Figure 5-17. The possibilities for generic highlights extraction based on affective content modeling

Finally, let us see on the example of a sport video sequence from Section 5.4.3 (Figure 5-7) how the highlights time curve is related to the original (non-filtered) arousal time curve, and how useful the highlights time curve is for highlights extraction. We first consider the case of M=3, where we are maximally selective when creating the highlighting video abstract, that is, we are interested in extracting the strongest (richest) highlights only. The highlights time curve for M=3 is shown in Figure 5-18b together with the original (non-filtered) arousal time curve $A(k)$. If we look at content labels characterizing different video segments in Figure 5-18a, we can see that the highlights time curve in Figure 5-18b provides highly distinguishable peaks at video segments corresponding to goals. Obviously, the goals appear to be the only events in a soccer TV broadcast complying with the requirements that we posed on the strength of the highlights in this case. The horizontal line in Figure 5-18b provides an abstract of 50 seconds showing the only two goals contained in the analyzed excerpt, each of them preceded by the action leading to the goal and succeeded by a number of shots showing the situation in the stadium, as well as by replays of the action taken from

different camera angles. The effect of filtering becomes clear when we move the cut-off line vertically: depending on the line position, only the length of the extracted video segments will change, but not the composition of the resulting highlighting video abstract, as it will always consist of the goals and the actions related to them only, while all other events of the game will be left out.

Figures 5-18c and 5-18d show the highlights time curves obtained using the same procedure as above but with weaker requirements posed on the strength of the highlights to be extracted, namely, with M=2 and M=1, respectively. Clearly, each reduction of the value of M resulted in an extension of the scope of extractable video content. In Figure 5-18c, besides the goals also a goal chance (free kick) and an action resulting in a game break (foul play) are now considered in the highlights extraction procedure. In Figure 5-18d, no further events are added to the highlights extraction base, except that more material is considered related to the "free kick" event between frames 6000 and 8000. This was also expected as in this video sequence no additional "exciting" events could be found. However, although the number of the extractable events is similar for M=2 and M=1, the ratio of the material extracted from different events will vary in both cases. As the peaks of the middle two events in Figure 5-18c are lower than the peaks of the goals, much longer segments will be extracted for the goals relatively to other two events. This is not the case in Figure 5-18d, where the peaks of all four events have almost equal height.

5.5.3 Personalized video delivery based on affective content extraction

The information on the affective video content can be used to enhance the quality of personalizing the video delivery to the user. The user may namely like or dislike a program largely depending on the prevailing mood of that program. We envision in this section two personalization scenarios that are based on this information.

5.5.3.1 Personalization based on the affective user-profile generation

In this scenario we assume that video is delivered to the user through a home video storage system. This system is not only capable of storing a large number of video hours, but also of processing and analyzing the stored digital video data, and of learning from its interaction with the user. We distinguish between two phases of this scenario, the *profile-learning* and the *profile-matching* phase.

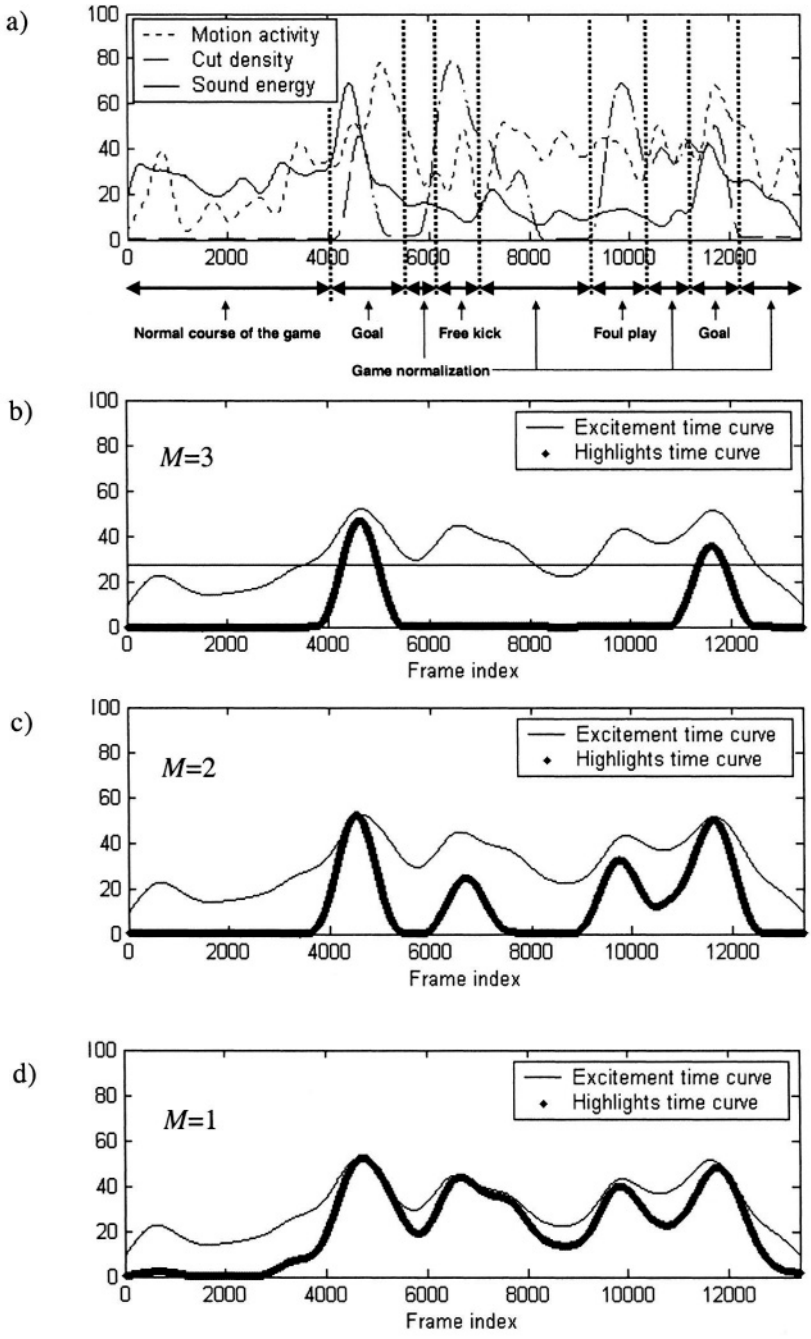

Figure 5-18. Adaptive highlights extraction

Profile learning

In the first phase a user profile is generated on the basis of previous program selections of the user. Here, the user is allowed to freely select the programs to watch over a given period of time. For each of the selected programs the affect curve and its gravity point are computed. This will result in a large number of gravity points scattered across the 2D affect space. User preferences will typically result in a number of clusters of gravity points. Each cluster can be seen as an "area of interest". In general, several areas of interest may emerge, as illustrated in Figure 5-19.

The obtained set of areas of interest can be seen as the affective profile of the user, as it implicitly represents user's preferences with respect to the prevailing mood of a video he or she likes most. This profile is to be distinguished from the "classical" profile of the user, that generally consists of facts, such as the number of times a program has been selected (non-semantic profile), or the list of preferred topics (semantic profile, e.g. in the case of news archive) [Boa03].

With any change of the preferences, the user will start selecting the programs again. This will mark the start of a new profile-learning phase. The change in user's preferences can best be seen through the shifting of the areas of interest after a considerable number of new programs have been selected, and after old areas of interest disappear that were not covered by newly selected programs to a substantial extent.

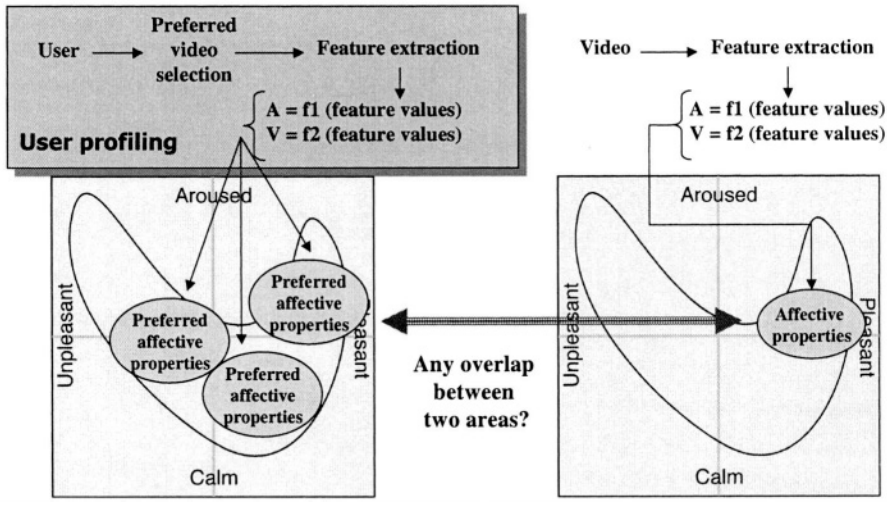

Figure 5-19. Personalized video delivery based on the affective profile of the user

Profile matching

Once the affective profile of the user has been generated, new programs can be delivered to the user based on the degree of matching between the prevailing moods of these programs and those belonging to the areas of interest. The prevailing mood of each incoming program is obtained again by computing the corresponding affect curve and its gravity point. The program is delivered to the user if its gravity point is within an area of interest.

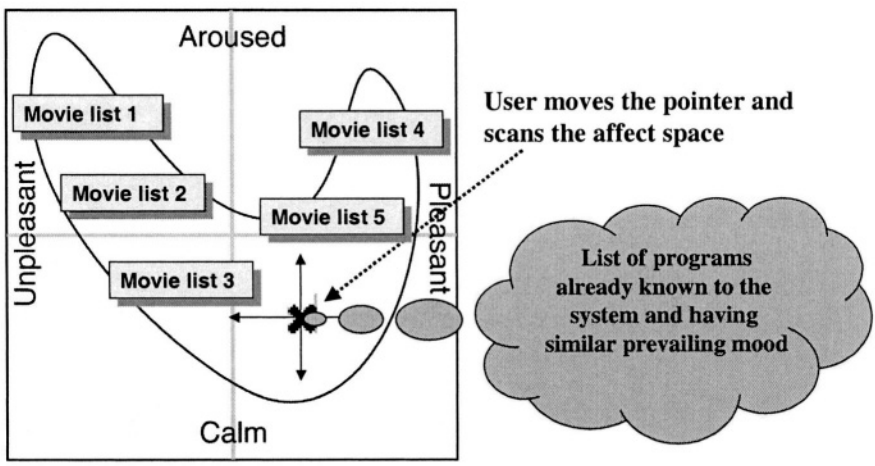

Figure 5-20. Personalized delivery based on affective browsing

5.5.3.2 Personalization through browsing the 2D affect space

An alternative to generating an affective profile of the user is to simply let the user browse through the 2D affect space. As illustrated in Figure 5-20, the user can use the remote control to move a pointer across the parabolic surface. As the labels "calm" and "aroused" for the arousal and "pleasant' versus "unpleasant" for the valence are meaningful to the user, a first selection of the program types to be downloaded in the future can be done rather easily. For instance, the user moves to the left for horrors and thrillers, and to the right for comedies. By moving the pointer up or down on the 2D affect space the preference for more or less exciting programs, respectively, can be specified.

An additional clue securing the proper specification of the preferences are the program lists appearing automatically at each pointer location. These lists include all programs that the user has already seen before and that have

the prevailing mood corresponding to the affective state of that location. In this way, the user may check whether he or she is in the right part of the 2D affect space simply by looking at the programs on each list: "Do I want to have more programs like these with respect to the prevailing mood?".

The advantage of this scenario as compared to the previous one, is that no learning is required. Personalized video delivery can start immediately, while the confidence in proper selection of preferences will grow with more and more programs being viewed and included in the program lists.

5.6 REMARKS AND RECOMMENDATIONS

Compared to the topics discussed in previous chapters of this book, the topic of affective video content analysis is definitely least developed. This is not surprising in view of high difficulty of the problem addressed here: extracting emotions, feelings and moods from sounds and pictures. It is, however, clear that research in this direction is necessary in order to provide the theoretical and algorithmic basis for realizing many important applications, which could not be possible by analyzing video at the cognitive level alone.

While the framework for analyzing a video at the affective level is reasonably well defined by the valence and arousal axes and the 2D affect space, little is known about the possibilities to map the low-level information extracted from video (features) onto the points in the affect space. Although a large number of audio-visual features have already been related in one way or another to affective dimensions, these relations are, however, rather vague and, therefore, difficult to be employed in the development of models for the arousal and valence time curves. Clearly, a reliable feature pool is needed to provide the basis for inferring the values of arousal and valence from video data. Once this pool is available, models for arousal and valence time curves need to be developed that optimally exploit the information found in different modalities of video and realistically depict the changes in the type and intensity of human affective states in time, as evoked by the varying video content. The development of these models can be approached, for instance, via primitives, as we showed on the example of arousal and valence modeling in sections 5.4.3 and 5.4.4. Although the models used as examples in this chapter are rather simple, the presented results are promising and can serve as an inspiration for further research in this direction. More insight into the challenges of affective video content analysis and the theory (psychology, psychophysiology) that may be useful for meeting these challenges can be gained from the selected literature listed below.

5.7 REFERENCES AND FURTHER READING

[Ada00] Adams, B., Dorai, C., Venkatesh, S.: *Novel approach to determining tempo and dramatic story sections in motion pictures,* Proceedings of ICIP 2000, Vol. II, Vancouver 2000, pp 283-286

[Arn83] Arnheim, R.: *Film as art.* London: Faber & Faber, 1958/1983

[Bab03] Babaguchi N., Nitta N.: *Intermodal collaboration: a strategy for semantic content analysis for broadcasted sports video,* Proc. of IEEE ICIP 2003

[Boa03] Boavida M., Cabaco S., Correia N.: *A system for delivering personalized video content,* OOIS 2003 Workshop on Metadata and Adaptability in Web-based Information Systems, Geneve CH, September 2003

[Bor01] Bordwell D., Thompson K.: *Film Art: An Introduction,* McGraw-Hill, New York, 2001

[Bra91] Bradley, M. M. and Lang, P. J.: *International affective digitized sounds (IADS): technical manual and affective ratings.* Gainesville, University of Florida, Center for Research in Psychophysiology, 1991

[Bra94] Bradley, M.: *Emotional Memory: A dimensional analysis.* In Emotions: Essays on emotion theory, Hillsdale, NJ: LEA, 1994

[Cha96] Chang Y.-L., Zeng W, Kamel I., Alonso, R.: *Integrated image and speech analysis for content-based video indexing,* Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems, 1996, Page(s): 306–313

[Chu95] Chung S.-J.: *An acoustic and perceptual study on the emotive speech in Korean and French,* in ICPhS, Vol. 1, Session 11.7, pp. 266-269, Stockholm 1995

[Col99] Colombo C., Del Bimbo A., Pala P.: *Semantics in Visual Information Retrieval,* IEEE Multimedia, July-September 1999, pp. 38-53

[Dag01] Dagtas S., Abdel-Mottaleb M.: *Extraction of TV highlights using multimedia features,* IEEE Fourth Workshop on Multimedia Signal Processing, 2001, Page(s): 91 -96

[Dav64] Davitz J.R.: *The communication of emotional meaning,* McGraw-Hill Book Company, New York, 1964

[Det97] Detenber, B.H., Simons, R.F., Bennett, G.G.: *Roll 'em!: The effects of picture motion on emotional responses.* Journal of Broadcasting and Electronic Media, 21, 1997, pp 112-126

[Det00] Detenber B.H.: *The emotional significance of color in television presentations,* Mediapsychology, 2, pp 331-355, 2000

[Die99] Dietz, R., Lang, A.: Aefective Agents: *Effects of Agent Affect on Arousal, Attention, Liking and Learning,* 3rd International Cognitive Technology Conference, CT'99, 1999

[Eki03] Ekin A., Tekalp A.M.: *Robust dominant color region detection and color-based applications for sports video,* Proc. of IEEE ICIP 2003

[Fit92] Fitzgibbons, L., Simmons R.F.: *Affective response to color-slide stimuli in subjects with physical anhedonia: A three-systems analysis,* Psychophysiology, 29(6), pp. 613-620, 1992

[Gar85] Gardner M.P.: *Mood states and consumer behavior: A critical review,* Journal of Consumer Research, 12, pp. 281-300, December 1985

[Gia76] Giannetti, L. D.: *Understanding movies* (second edition). Englewood Cliffs, NJ: Prentice-Hall, 1976

[Gon95] Gong Y., Sin L.T., Chuan C.H., Zhang H., Sakauchi M.: *Automatic Parsing of TV Soccer Programs,* ICMCS '95, pp. 167 –174, May 1995

[Gra98] Grand, S., Cliff, D.: *Creatures: Entertainment software agents with artificial life,* Autonomous Agents and Multi-Agent Systems, 1(1), 1998, pp 39-57

[Gre89] Greenwald, M. K., Cook, E. W., Lang, P. J.: *Affective judgment and psychophysiological response: Dimensional covariation in the evaluation of pictorial stimuli.* Journal of Psychophysiology, 3, 1989, pp 51-64

[Haa88] Haas C.R.: *Advertising Practice (Pratique de la Publicité),* Bordas, Paris, 1988 (In French)

[Han01] Hanjalic A., Xu. L.-Q.: *User-oriented affective video content analysis,* IEEE Workshop on Content-Based Access of Image and Video Libraries 2001 (CBAIVL 2001), pp. 50 –57, December 2001

[Han03] Hanjalic A.: *Generic approach to highlights extraction from a sport video,* IEEE International Conference on Image Processing, Barcelona, 2003

[Han04] Hanjalic A. Xu L.-Q.: *Affective Video Content Representation and Modeling,* IEEE Transactions on Multimedia, 2004, to appear

[Hop94] Hopkins R., Fletcher J.E.: *Electrodermal measurement: Particularly effective for forecasting message influence on sales appeal,* in A. Lang (Eds.): Measuring psychological responses to media, pp. 113-132, Hillsdale NJ, Lawrence Erlbaum Associates, 1994

[Hua02] Hua W., Han M., Gong Y.: *Baseball scene classification using multimedia features.* Proceedings of IEEE International Conference on Multimedia and Expo, 2002. ICME '02. Volume: 1 , 26-29 Aug. 2002, Page(s): 821 -824 vol.1

[Jai02] Jaimes A., Echigo T., Teraguchi M., Satoh F.: *Learning personalized video highlights from detailed MPEG-7 metadata,* IEEE International Conference on Image Processing (ICIP), Volume 1, 2002

[Kaw98] Kawashima, T.; Tateyama, K.; Iijima, T.; Aoki, Y.; *Indexing of baseball telecast for content-based video retrieval* Proceedings of International Conference on Image Processing (ICIP), Volume: 1 , 4-7 Oct. 1998 Page(s): 871-874 vol.1

[Lan80] Lang P.J.: *Behavioral treatment and bio-behavioral treatment: Computer applications,* in J.B. Sidowski, J.H. Johnson and T.A. Williams (Eds.): Technology in mental health care delivery systems, pp.119-137, Norwood NJ: Ablex, 1980

[Lan95a] Lang, P. J.: *The network model of emotion: Motivational connections.* In R. S. Wyer & T. K. Srull (Eds.), Advances in social cognition (Vol. 6). Hillsdale, NJ: Lawrence Erlbaum Associates, 1995

[Lan95b] Lang, A. Dhillon, P., Dong, Q: *Arousal, Emotion, and Memory for television messages.* Journal of Broadcasting and Electronic Media, 38, 1995, pp 1-15

[Lan85] Lang, P. J. and Greenwald, M. K.: *The international affective picture system slides and technical report.* Gainesville, University of Florida, Center for Research in Psychophysiology, 1985

[Lan96] Lang, A., Newhagen, J., Reeves, B: *Negative Video as Structure: Emotion, attention, capacity, and memory.* Journal of Broadcasting and Electronic Media, 40, 1996, pp 460-477

[Les97] Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., Bhoga, R.S.: The persona effect: affective impact of animated pedagogical agents, in Proc. Human Factors Comput. Syst., 1997, pp. 359—366

[Leo03] Leonardi R., Megliorati P., Prandini M.: *Semantic indexing of sports program sequences by audio-visual analysis,* Proc. of IEEE International Conference on Image Processing (ICIP), 2003

[Li01] Li, B.; Sezan, M.I.: *Event detection and summarization in sports video,* IEEE Workshop on Content-Based Access of Image and Video Libraries, 2001. (CBAIVL), Page(s): 132-138

[Li03] Li B., Sezan M.I.: *Semantic sports video analysis: approaches and new applications,* Proc. of IEEE International Conference on Image Processing (ICIP), 2003

[Mic96] Microsoft, *IntelliSense in Microsoft Office 97.* Microsoft Office 97 Whitepaper, 1996

[Mon81] Monaco, J.: *How to Read a Film: The Art, Technology, Language, History and Theory of Film and Media,* Oxford University Press, 1981

[Mur93] Murray I.R., Arnott J.L: *Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion,* Journal of Acoustical Society of America, 93 (2), pp. 1097-1108, February 1993

[Nas94] Nass, C., Steuer, J. and Tauber, E. R.: *Computers Are Social Actors,* in Proceedings of CHI '94, Human Factors in Computing Systems. ACM Press, 1994, pp 72-78

[Nit00] Nitta, N., Babaguchi, N., Kitahashi, T.: *Extracting actors, actions and events from sports video - a fundamental approach to story tracking* Proceedings. 15th International Conference on Pattern Recognition, (ICPR), Volume: 4 , Page(s): 718-721

[Osg57] Osgood, C., Suci, G., Tannenbaum, P.: *The measurement of meaning.* Urbana, IL: University of Illinois Press, 1957

[Pan01] Pan H., van Beek P., Sezan M.I.: *Detection of slow-motion replay segments in sports video for highlights generation,* Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2001

[Pet02] Petkovic, M., Mihajlovic, V., Jonker, W., Djordjevic-Kajan, S.: *Multi-modal extraction of highlights from TV Formula 1 programs,* Proceedings of the IEEE International Conference on Multimedia and Expo 2002 (ICME), Volume: 1, Page(s): 817-820 vol.1

[Pic97a] Picard, R.: *Affective Computing,* MIT Press, 1997

[Pic97b] Picard, R., Cosier, G.: *Affective Intelligence – The Missing Link?,* BT Technology Journal, Vol.14, No.4, October 1997, pp 150-161

[Pit90] Pittam J., Gallois C., Callan V.: *The long-term spectrum and perceived emotion,* Speech Communication, 9, pp. 177-187, 1990

[Rom95] Romer D.: *The Kodak picture exchange,* Seminar at MIT Media Lab, April 1995

[Rui00] Rui Y., Gupta A., Acero A.: *Automatically extracting highlights for TV baseball programs,* Proc. ACM Multimedia 2000, Los Angeles CA, 2000

[Rus77] Russell, J., & Mehrabian, A.: *Evidence for a three-factor theory of emotions.* Journal of Research in Personality, 11, 1977, pp 273-294

[Sch81] Scherer K.R.: *Speech and emotional states.* Chapter 10 in J.K. Darby (Eds.) Speech evaluation in psychiatry, pp. 189-220, Grune and Stratton Inc., 1981

[Sch54] Schlosberg H.: *Three dimensions of emotion,* Psychological Review, 61(2): 81-88, March 1954

[Sim99] Simons R., Detenber B.H., Roedema T.M., Reiss J.E.: *Emotion-processing in three systems: The medium and the message,* Psychophysiology, 36, pp. 619-627, 1999

[Sno03] Snoek, C.G.M., Worring, M.: *Time interval maximum entropy based event indexing in soccer video* Proceedings of the IEEE International Conference on Multimedia and Expo 2003 (ICME), Volume: 3 , Page(s): 481 -484

[Sud98] Sudhir, G., Lee, J.C.M., Jain, A.K.: *Automatic classification of tennis video for high-level content-based retrieval,* Proceedings of the IEEE International Workshop on Content-Based Access of Image and Video Database 1998., Page(s): 81 -90

[Tos96] Tosa N., Nakatsu R.: *Life-like communication agent – emotion sensing character 'MIC' and feeling session character 'MUSE',* in Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS), pp. 12-19, 1996

[Uts02] Utsumi, O., Miura, K., Ide, I., Sakai, S., Tanaka, H.: *An object detection method for describing soccer games from video,* Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Volume: 1 , 26-29 Aug. 2002 Page(s): 45-48 vol.1

[Wil69] Williams C.E., Stevens K.N.: *On determining the emotional state of pilots during flight: An exploratory study,* Aerospace Medicine, 40(12): pp. 1369-1372, December 1969

[Wil72] Williams C.E., Stevens K.N.: *Emotions and Speech: Some acoustical correlates,* Journal of Acoustical Society of America, 52(4): pp. 1238-1250, 1972, part 2

[Xie02] Xie L., Chang S.-F., Divakaran, A., Sun H.: *Structure analysis of soccer video with hidden Markov models,* Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2002 (ICASSP), Volume: 4 , Page(s): IV-4096 -IV-4099 vol.4

[Xio03] Xiong, Z., Radhakrishnan, R., Divakaran, A., Huang, T.S.: *Audio Events Detection Based Highlights Extraction from Baseball, Golf and Soccer Games in a Unified Framework,* IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol. 5, pp. 632-635, 2003

[Xu03] Xu G., Ma Y.-F., Zhang H.-J., Yang S.: *A HMM based semantic analysis framework for sports game event detection,* Proceedings of IEEE International Conference on Image Processing (ICIP), 2003

Index